**ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**Họ và tên sinh viên : Thiều Trọng Bảo**

**BÁO CÁO**

**THỰC TẬP TỐT NGHIỆP**

**Công ty thực tập :WATA solutions**

**Chuyên gia hướng dẫn :Vương Chí Quang**

**Giảng viên hướng dẫn :Nguyễn Quốc Huy**

***TP. Hồ Chí Minh, tháng 09 năm 2020***

**MỤC LỤC**

[NHẬN XÉT CỦA CHUYÊN GIA DOANH NGHIỆP 4](#_Toc49651007)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 5](#_Toc49651008)

[LỜI MỞ ĐẦU 6](#_Toc49651009)

[CHƯƠNG 1. GIỚI THIỆU 7](#_Toc49651010)

[1.1. GIỚI THIỆU CÔNG TY THỰC TẬP 7](#_Toc49651011)

[1.2. NHIỆM VỤ THỰC TẬP 12](#_Toc49651012)

[KẾT LUẬN CHƯƠNG 1 12](#_Toc49651013)

[CHƯƠNG 2. QUÁ TRÌNH THỰC TẬP 13](#_Toc49651014)

[2.1. TỔNG QUAN VÀ CÁCH QUẢN LÍ DỰ ÁN 13](#_Toc49651015)

[2.2. CÀI ĐẶT MÔI TRƯỜNG 16](#_Toc49651020)

[2.3. GIỚI THIỆU SPRING FRAMEWORK, SPRING BOOT 19](#_Toc49651026)

[2.4. SỬ DỤNG POSTMAN 21](#_Toc49651031)

[2.5. TÌM HIỂU SPRING ANNOTATION 26](#_Toc49651032)

[2.6. TÌM HIỂU SPRING CONTROLLER 29](#_Toc49651033)

[2.7. TÌM HIỂU SPRING DATA 31](#_Toc49651034)

[2.8. TÌM HIỂU SPRING SECURITY 33](#_Toc49651035)

[2.9. KẾT QUẢ THỰC NGHIỆM 38](#_Toc49651040)

[KẾT LUẬN CHƯƠNG 2 39](#_Toc49651041)

[CHƯƠNG 3: KẾT QUẢ THỰC TẬP 40](#_Toc49651042)

[CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ 44](#_Toc49651043)

[TÀI LIỆU THAM KHẢO 45](#_Toc49651044)

[PHỤ LỤC 46](#_Toc49651045)

# NHẬN XÉT CỦA CHUYÊN GIA DOANH NGHIỆP

......................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................... ................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................... ........................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................... ................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................... ...................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

# LỜI MỞ ĐẦU

Ngành Công nghệ thông tin hiện nay là một trong những ngành mũi nhọn không chỉ ở Việt Nam mà ở trên toàn thế giới. Mọi lĩnh vực trong đời sống đều có thể ứng dụng Công nghệ thông tin, từ kinh doanh đến tài chính, từ quản lý một cửa hàng nhỏ đến quản lý cả một hệ thống trực thuộc chính phủ. Một đất nước không có Công nghệ thông tin sẽ rất khó phát triển vì nó hầu như có thể giải quyết các vấn đề một cách nhanh chóng và chính xác, giúp tiết kiệm thời gian của con người cũng như giúp mọi thứ thuận tiện và dễ dàng hơn. Hiểu được điều đó, em chọn ngành Công nghệ thông tin để học tập và định hướng công việc sau này.Ngày nay, khi mà Internet càng ngày càng phát triển, thì các doanh nghiệp cũng dần chuyển các hoạt động của mình lên nền tảng Website, trong đó nổi bật lên việc xây dựng API. Đây là một hướng đi tuyệt vời cho tương lai vì những ưu điểm tuyệt vời về hiệu năng mà xu hướng này mang lại. Đồng thời em xin cảm ơn Tiến sĩ Nguyễn Quốc Huy – giảng viên trường Đại học Sài Gòn khoa Công nghệ thông tin, công ty WATA SOLUTIONS đã tạo điều kiện cho em có cơ hội thực tập tại môi trường làm việc thật sự, qua đó gặt hái được những kinh nghiệm và kỹ năng quý báu song hành với những kiến thức đã được học ở trường, cũng như tận tình giúp đỡ em hoàn thành báo cáo thực tập này. Dĩ nhiên sẽ không thể tránh khỏi sai sót nhưng em sẽ cố gắng cải thiện và tiếp thu ý kiến từ mọi người để nâng cao giá trị bản thân, trở thành một người có ích cho xã hội nói chung và ngành Công nghệ thông tin nói riêng.

# CHƯƠNG 1. GIỚI THIỆU

## GIỚI THIỆU CÔNG TY THỰC TẬP

**Tên công ty** : WATA Solutions

**Địa chỉ:** 247/8B Hoàng Hoa Thám, Phường 05, Quận Phú Nhuận, TP Hồ Chí Minh

**Email công ty**: [info@watasolutions.com](mailto:info@watasolutions.com)

**Website công ty** : watasolutions.com

**Telephone**: 028 3636 9248

**Địa chỉ công ty trên các mạng xã hội**:

Linkedin: <https://www.linkedin.com/company/wata-solutions-ltd/about/>

Facebook: <https://www.facebook.com/watasolutions/>

**Thời gian làm việc**: Thứ 2 – Thứ 6 (9h – 18h00), nghỉ trưa 12h00 – 13h00

**Lịch sử phát triển của công ty :**

Tiền thân được thành lập vào ngày 02/03/2015, tọa lạc tại Tòa Nhà SBI, Lô 6B, Đường Số 3, Công Viên Phần Mềm Quang Trung, Phường Tân Chánh Hiệp, Quận 12, Hồ Chí Minh, Vietnam.

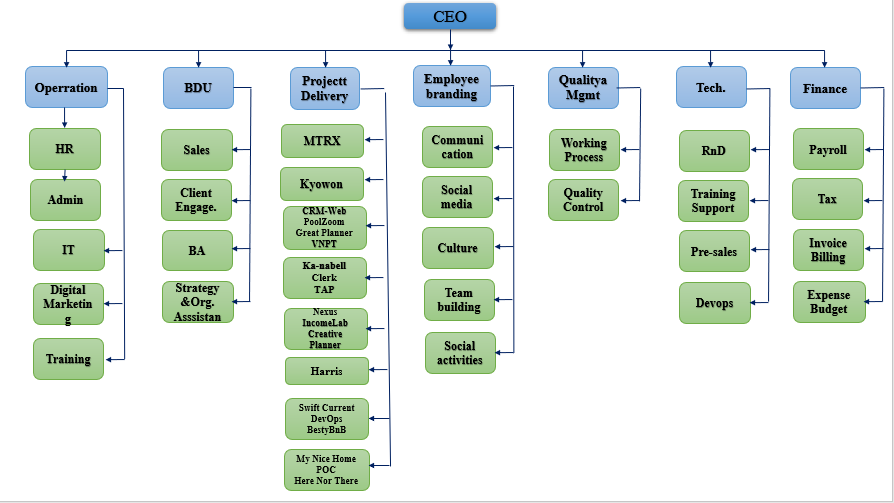
Với sự phát triển không ngừng của mình , công ty dần mở rộng phạm vi của mình và hiện tại đã có đến 4 văn phòng và khoảng 100 nhân viên. Đến năm 2020, công ty chuyển văn phòng từ Công viên phần mềm Quang Trung đến 247/8B Hoàng Hoa Thám, Phường 05, Quận Phú Nhuận, TP Hồ Chí Minh.

Với sự phát triển và đổi mới không ngừng , công ty hiện có nhiều đối tác trên khắp thế giới từ Mỹ , Hàn, Nhật , Trung cũng như nhiều nước khác,….

**Cơ sở vật chất của doanh nghiệp:** được trang bị máy tính để bàn, màn hình, máy in, phòng họp, phòng giải trí, nơi ăn uống.



**Cơ cấu tổ chức của doanh nghiệp:** Hiện tại có 4 văn phòng lần lượt được đặt tại Việt Nam 3 văn phòng và 1 văn phòng ở Singapore



**Lĩnh vực hoạt động của doanh nghiệp:**

**Mobile Application:** Cung cấp các dịch vụ liên quan đến phát triển mobile như:

1. Mobile Applications
2. Mobile E - Commerce
3. Responsive Design

**Website:** Cung cấp các dịch vụ liên quan đến phát triển website như:

1. Web design : Luôn cố gắng tạo ra một ứng dụng hấp dẫn nhằm tối đa và tối ưu hóa chức năng giải trí và kinh doanh
2. Web business: Xây dựng sản phẩm cho phép trên nhiều thiết bị khác nhau mà không cần phải duy trì nhiều phiên bản web riêng biệt.
3. Web E – Commerce: Xây dựng ứng dụng trực quan, dễ điều hướng để giúp khách hàng dễ dàng thanh toán , dù họ ở bất cứ đâu.
4. Web Application: Xây dựng ứng dụng web có thể sử dụng trên nhiều thiết bị khác nhau.

**UI/UX design:** Nhằm tạo ra một UI/UX tốt nhất, WATA luôn luôn lắng nghe khách hàng, chỉ hài lòng khi khách hàng hài lòng. Từ lúc bắt đầu một dự án, WATA sẽ luôn cố gắng hết mình để tạo ra và hoàn thành một sản phẩm mà khách hàng hài lòng và sẽ tiếp tục phát triển cho đến khi lấy được sự hài lòng đó từ khách hàng.

**Software Testing:** WATA cung cấp đầy đủ các dịch vụ kiểm thử bao gồm kiểm thử thủ công và kiểm thử tự động. Ngoài ra còn cung cấp, tư vấn các giải pháp kiểm thử khác nhau tùy thuộc vào yêu cầu của khách hàng

**Các đối tác hiện có của doanh nghiệp:** Wata Solutions làm việc với nhiều khách hàng trên khắp thế giới, cung cấp các dịch vụ và giải pháp phần mềm cho khách hàng trong nhiều lĩnh vực, bao gồm Công nghệ, Dịch vụ Tài chính, Thương mại Điện tử & Bán lẻ, Giáo dục và Chăm sóc sức khỏe.



Một số nhận xét về WATA Solutions của các đối tác trên thế giới

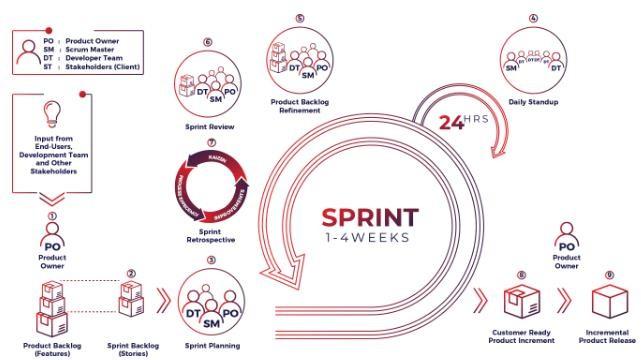
David Trice - CEO and Co-Founder of CX Technologies:’ The best mobile and multi-channel experience of any outsourcing vendor we interviewed. The choice was easy. Thanks so much.’

Jeremy Marcel CEO and Co-Founder of Syntax : ‘We liked the idea of the idea optimization and perfect of web design by WATA SOLUTIONS, cost-effective and high quality.’

Julia Franke CDO & Marketing DOJO: ‘ WATA SOLUTIONS by far, the best mobile and multi-channel experience of any outsourcing vendor we interviewed. The choice was easy.’

**Các quy trình của doanh nghiệp : Mô hình Scrum**

* Scrum là một Framework về quy trình và quản lý giúp giải quyết các vấn đề phức tạp, nhưng vẫn đảm bảo tính hiệu quả, sáng tạo và sản phẩm được tạo ra phải đạt được giá trị cao nhất. Bản thân Scrum là một Framework đơn giản nhằm giúp việc phối hợp hiệu quả nhất giữa các thành viên trong đội phát triển khi thực hiện những sản phẩm phức tạp. Với Scrum, sản phẩm được xây dựng trên 1 chuỗi các quy trình lặp lại (gọi là Sprint). Các sprint diễn ra đều đặn, mỗi một sprint là cơ hội để học hỏi điều chỉnh nhằm đạt được sự phù hợp và kết quả tốt nhất.
* Khi áp dụng Scrum, có 4 cuộc họp (Meetings or Ceremonies) quan trọng tạo nên cấu trúc trong mỗi Sprint như sau:
* Sprint planning: Cuộc họp lên kế hoạch của đội dự án, nhằm xác định những gì cần hoàn thành trong Spring sắp tới.
* Daily stand-up: Cũng được biết đến như “Daily Scrum”, một cuộc họp nhỏ 15 phút mỗi ngày để trao đổi công việc giữa đội phát triển.
* Sprint demo: Một cuộc họp chia sẻ, nơi mà các thành viên chỉ ra những gì họ đã làm được trong Sprint đó.
* Sprint retrospective: Sự đánh giá, nhìn lại những điều đã làm được và chưa làm được của Sprint hiện tại, và đưa ra giải pháp hành động cho Sprint tiếp theo được tốt và hoàn thiện hơn.
* Một nhóm phát triển Scrum thì sẽ có những thành phần hơi khác biệt so với mô hình truyền thống Waterfall, với 3 vai trò sau:
* Product Owner
* Scrum Master
* Development Team



Hình 5. Cấu trúc của team SCRUM

* Sinh viên được phân công thực tập ở bộ phận phát triển Website
* Thông tin ngắn về chuyên gia của doanh nghiệp hướng dẫn trực tiếp:
* Họ và tên: Vương Chí Quang
* Chức vụ: Team Leader – Trưởng nhóm dự án
* Email: quang.vuong@watasolutions.com
* Điện thoại: 0908 868 527
* Những nội dung mà sinh viên cần học hỏi thêm khi thực tập tại doanh nghiệp: quy trình xây dựng Single Page, những kỹ năng mềm cần có khi tham gia các dự án.

## Nhiệm vụ thực tập

Sinh viên tham gia vào project Payslip với vai trò back-end, CGDN cung cấp danh sách những kiến thức cần thiết tìm hiểu để nắm vững nền tảng trước khi bắt đầu vào project. Project Payslip thiết kế đăng nhập thông tin nhân viên, điểm danh và tính lương cho nhân viên.

Nghiên cứu công nghệ Spring framework.

## Kết luận chương 1

Công ty WATA Solutions là một doanh nghiệp tốt cho sinh viên thực tập vì đây được ứng dụng quy trình Agile nhằm đem lại sản phẩm tốt nhất cho thị trường. Sinh viên được tham gia vào các dự án thực tế để ứng dụng những gì đã được học ở trường vào môi trường doanh nghiệp, phát triển kỹ năng mềm của bản thân, học hỏi thêm nhiều kiến thức cũng như kinh nghiệm từ CGDN.

# CHƯƠNG 2. QUÁ TRÌNH THỰC TẬP

## 2.1. TỔNG QUAN VÀ CÁCH QUẢN LÍ DỰ ÁN

### **2.1.1. Tổng quan dự án**

* Tên dự án: Payslip
* Description: Quản lí nhân viên, các dự án và các task trong công ty. Ngoài ra còn dùng để chấm công cho nhân viên và một số chức năng khác.
* Ước lượng thời gian hoàn thành : 22/07/2020 – 11/07/2020
* Thành viên và vai trò trong dự án

|  |  |
| --- | --- |
| Tên thành viên | Vai trò trong dự án |
| Thiều Trọng Bảo | Back-end Dev (Spring Boot) |
| Nguyễn Thị Xuân Linh | Back-end Dev (Spring Boot) |
| Phùng Hà Thảo | Front-end Dev (Angular) |

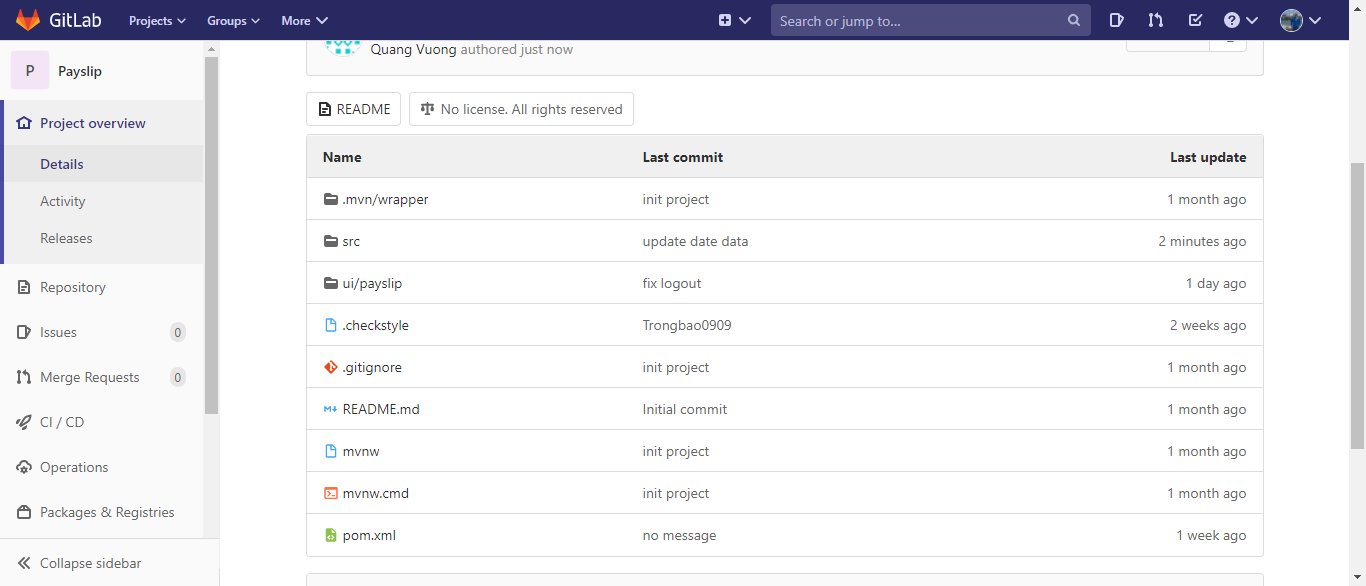
* Mentor/PM : Vương Chí Quang
* Quy trình phát triển dự án : Scrum
* Yêu cầu chức năng

|  |  |
| --- | --- |
| **Chức năng** | **Mô tả** |
| Đăng nhập | Cho phép đăng nhập nếu đúng thông tin. |
| Đăng xuất | Xóa Cookies người dùng |
| Quên mật khẩu | Kiểm tra và nhập mật khẩu mới |
| Quản lý nhân viên | Thêm, sửa, xóa, cập nhật |
| Quản lý project | Thêm, xửa, xóa, cập nhật, phân công |
| Quản lí tài khoản | Tạo mới, kich hoạt ,gửi mail |
| Xác thực | Xác nhận thông tin yêu cầu |
| Phân quyền | Gán vai trò tùy theo từng api |
| Quản lí bảng chấm công | Tính công cho nhân viên |
| Quản lí task | Chia task, phân công ,cập nhật |
| Bảo mật | Chặn những truy cập trái phép, Yêu cầu quyền truy cập mỗi khi truy cập |
| Ràng buộc dữ liệu | Kiểm tra dữ liệu input và output |
| Một số chức năng khác | Tìm kiếm, Phân trang, Xử lí dữ liệu. |

### **2.1.2. Quản lí dự án**

#### 2.1.2.1. GitLab

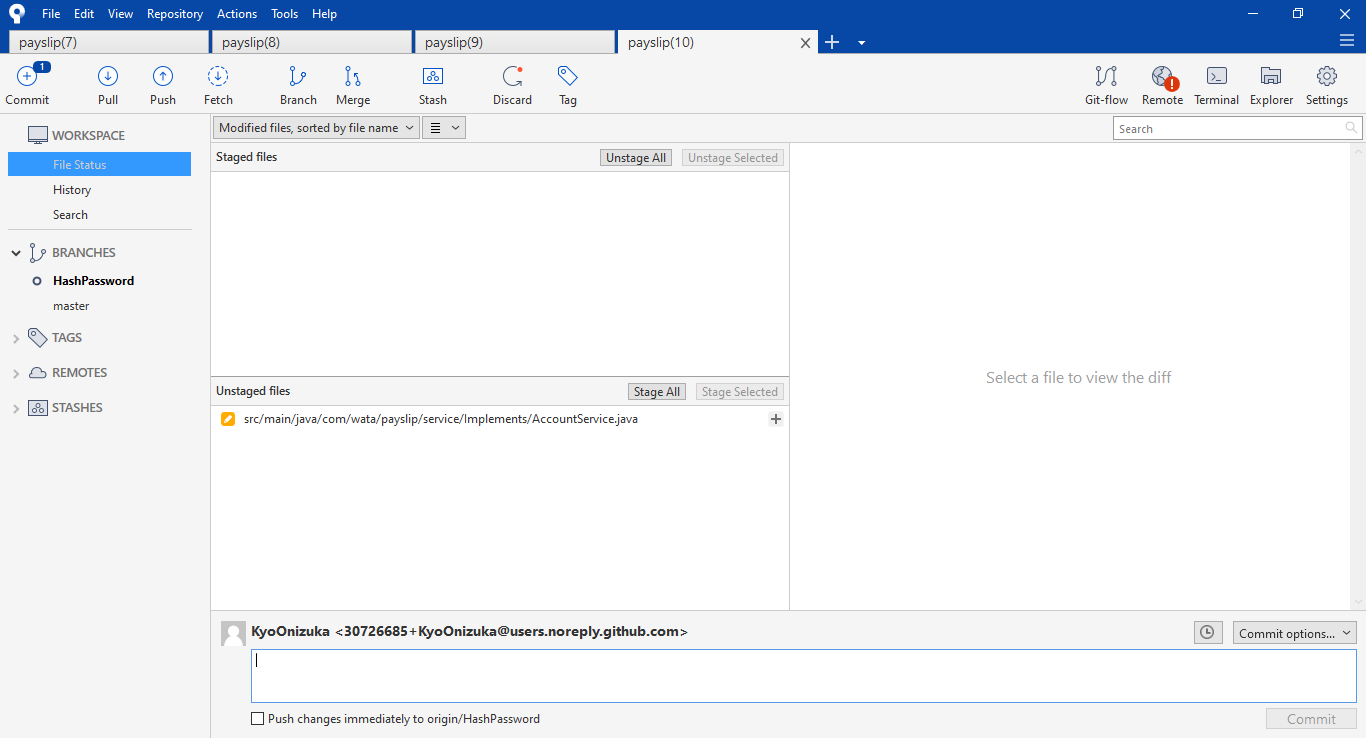
Giới thiệu: GitLab là hệ thống self-hosted mã nguồn mở dựa trên hệ thống máy chủ Git dùng để quản lý mã nguồn. GitLab cung cấp giải pháp server một cách hoàn hảo và nhận được sự đánh giá cao từ cộng đồng. Với **GitLab,** các tổ chức, cá nhân, doanh nghiệp có thể lưu trữ và quản lý kho code một cách khoa học, an toàn, truy cập nhanh chóng thông qua kết nối internet. GitLab cung cấp một dung lượng lưu trữ miễn phí cho người dùng và bạn có thể trả thêm phí nếu muốn tăng dung lượng.



Hình 1. Gitlab

#### 2.1.2.2. SourceTree

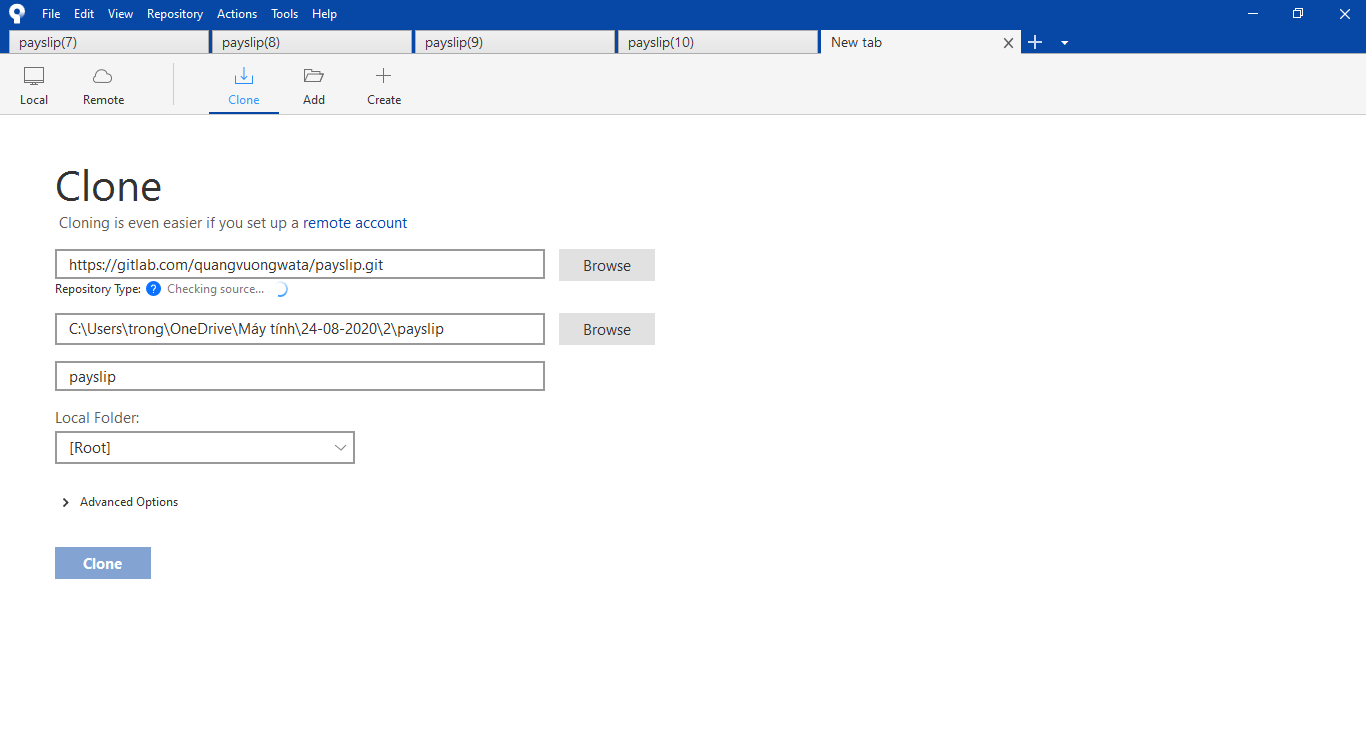
Giới thiệu: **SourceTree** là một ứng dụng hữu ích dành cho các developer .Dùng SourceTree để xem code trước khi commit sẽ giúp bạn tránh được code tạp nham, code thừa hay thiếu, hoặc xóa nhầm các thành phần không cần thiết.



Hình 2. Giao diện source tree

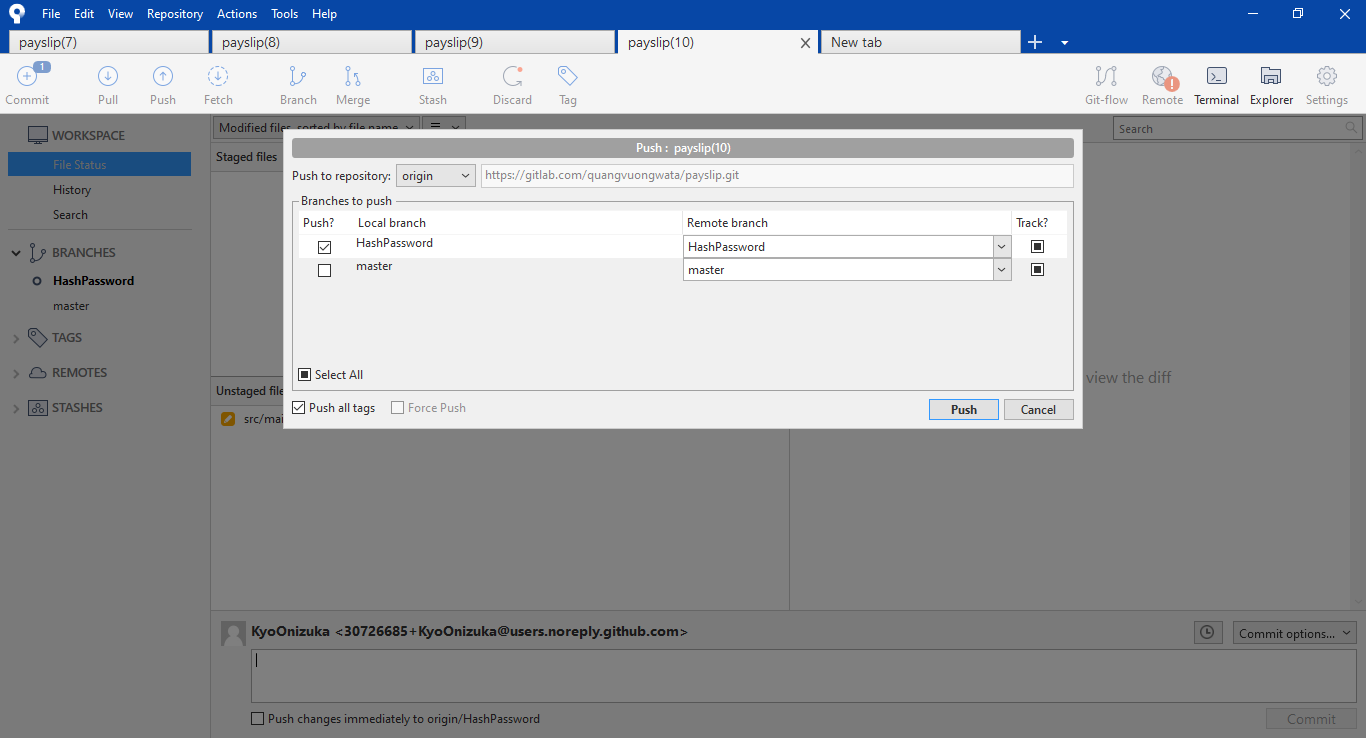
Các bước quản lí dự án với source tree

Bước 1: clone dự án từ Gitlab về



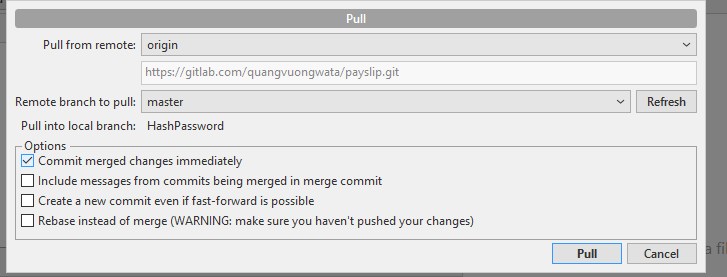
Hình 3. Clone project

Bước 2: Push dự án lên Gitlab

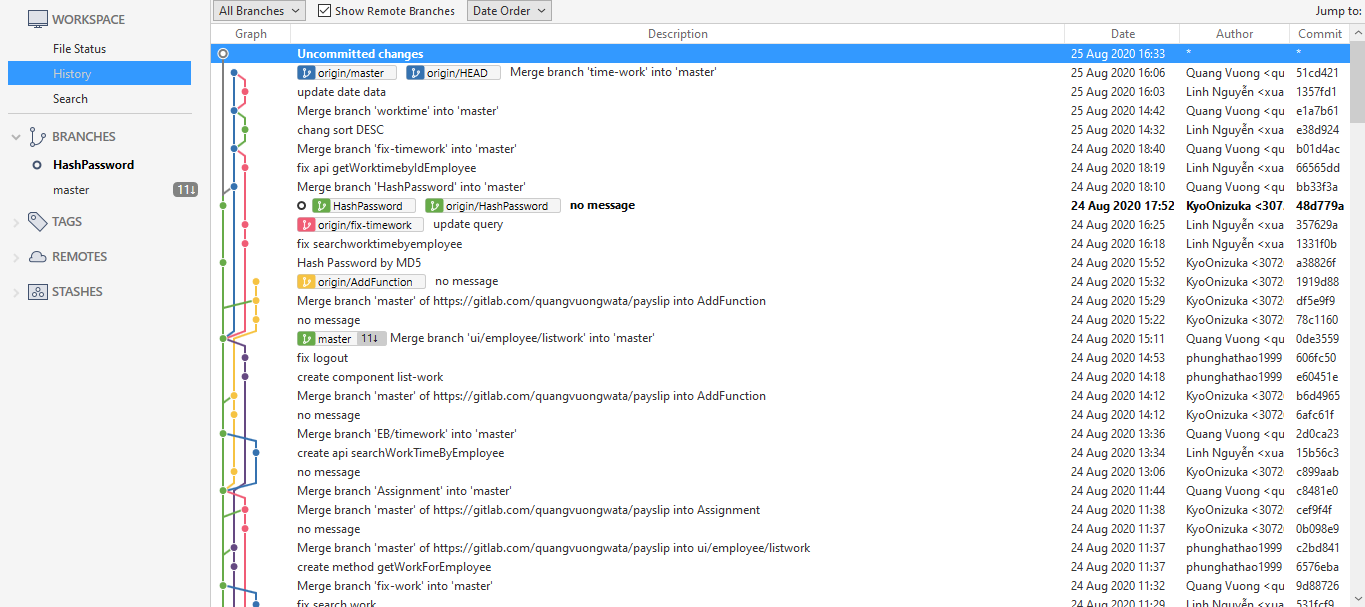


Hình 4. Push dự án lên gitlab

B3: Pull Code từ dự án trên GitLab



Hình 5. Pull dự án từ gitlab

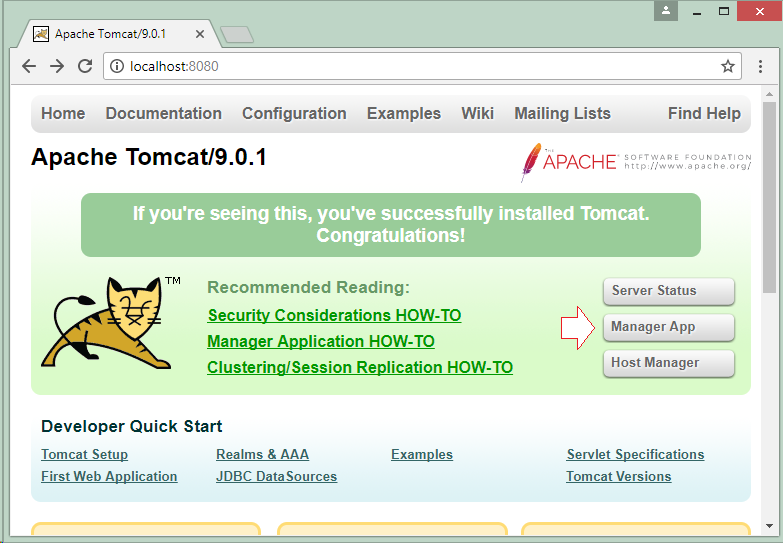


Hình 6. Sơ đồ biểu diễn các commit project được push lên gitlab

## 2.2. CÀI ĐẶT MÔI TRƯỜNG

### **2.2.1. Cài đặt Tomcat Server**

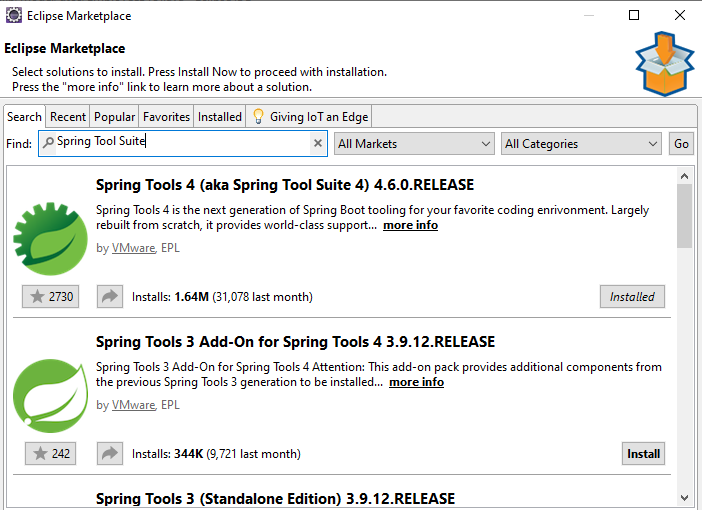
Sử dụng Tomcat Server để dùng làm server Backend



Hình 7. Giao diện Tomcat

### **2.2.2. Cài đặt Spring Tool Suite**

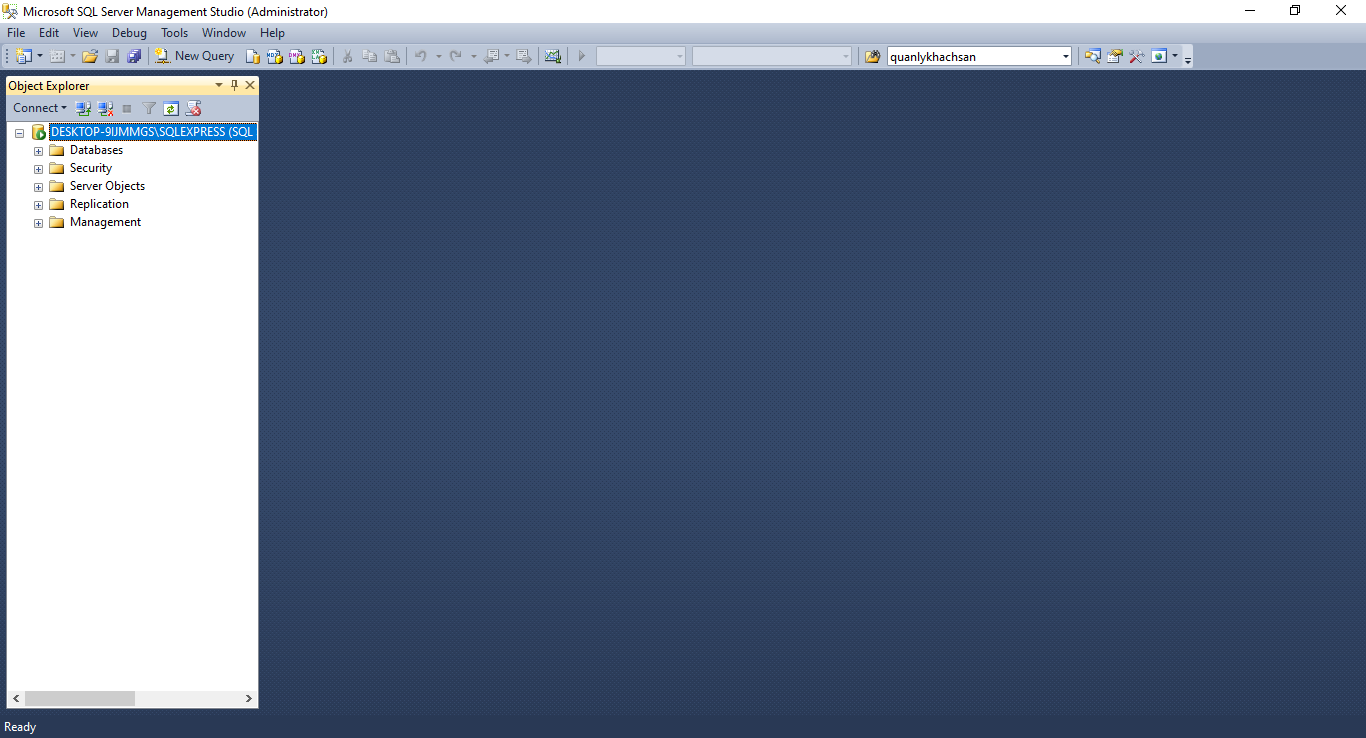
Vì dự án sử dụng Spring Boot nên chúng ta sẽ cần cài đặt Spring Tool Suite trong Eclipse



Hình 8. Spring Tool Suite

### **2.2.3. Cài đặt SqlServer**

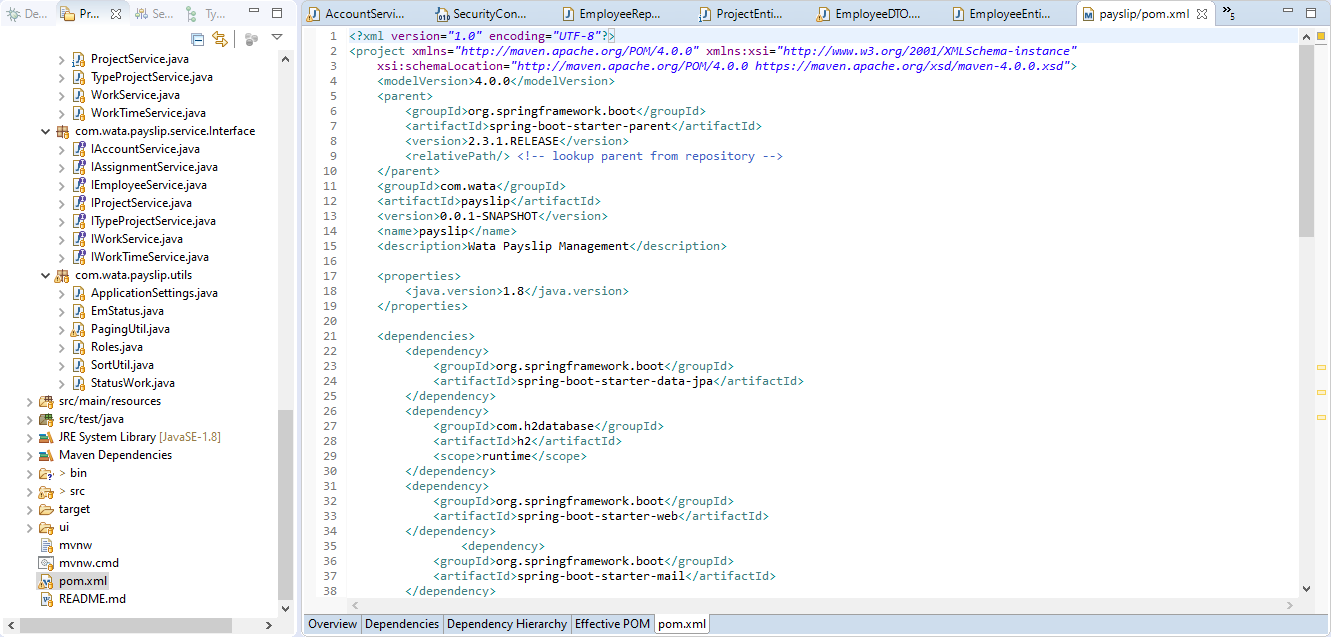
Trong dự án này, em sẽ chọn SqlServer để làm nới lưu trữ database bao gồm tất cả các dữ liệu sẽ sử dụng trong dự án.



Hình 9. MS SQL Server

### **2.2.4. Cấu hình file pom.xml**

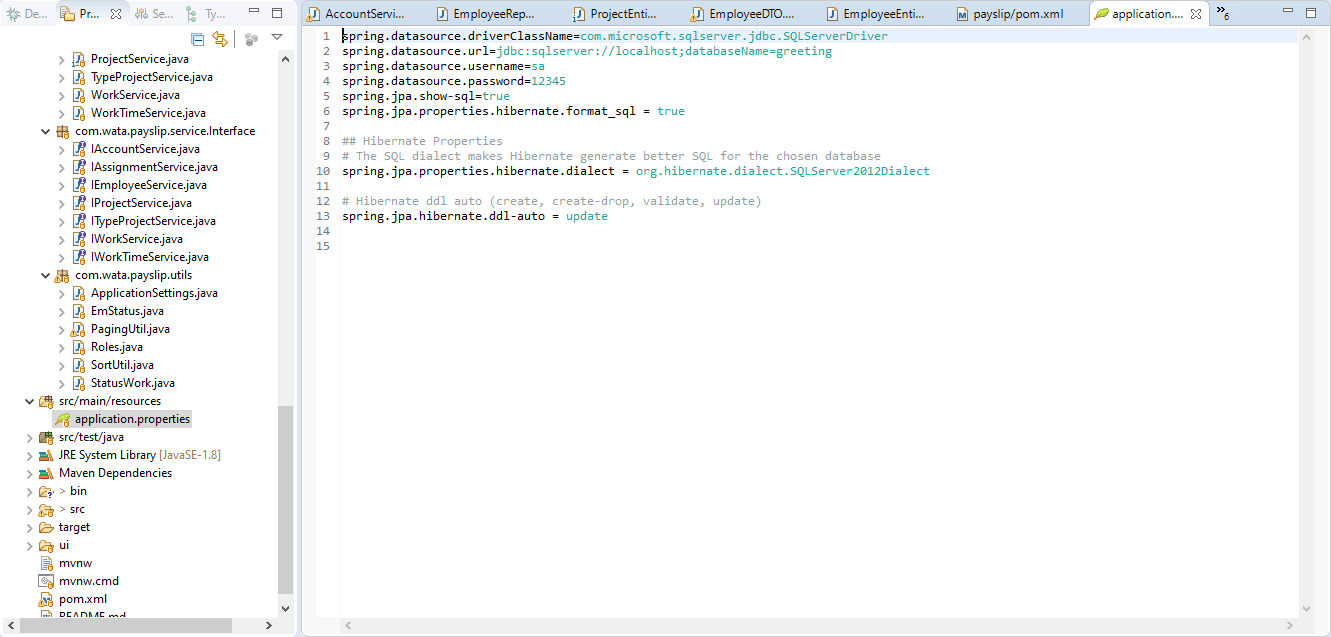
Trong quá trình xây dựng dự án, sẽ có nhiều tài nguyên chúng ta cần dùng tới, và chúng đều phải được khai báo trong file pom.xml chứ không phải tải theo cách thông thường.Khi đó thì những tài nguyên phù hợp sẽ được tự động tải về.



Hình 10. File pom.xml

### **2.2.5. Cấu hình driver jbdc để kết nối sqlserver**

Sau khi có database, chúng ta cần kết nối database với server. Vì vậy chúng ta có file cấu hình này, server sẽ tự động kết nối với database.

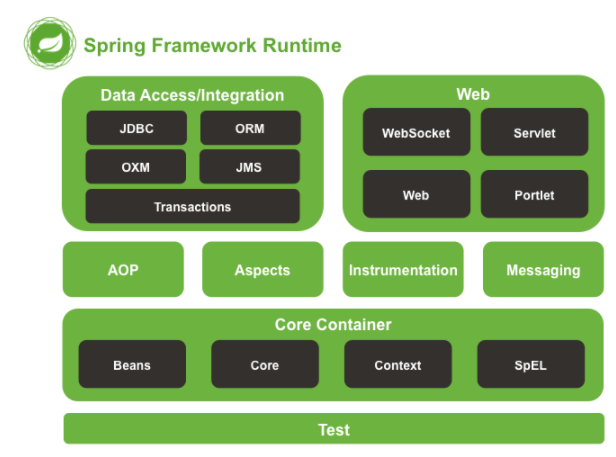


Hình 11. Kết nối database

## GIỚI THIỆU SPRING FRAMEWORK, SPRING BOOT

### **Spring là gì.**

* Spring là một Framework phát triển các ứng dụng java được các lập trình viên sử dụng. Nó giúp tạo các ứng dụng có hiệu năng cao, sử dụng lại code... Spring là mã nguồn mở, được phát triển, chia sẽ có cộng đồng người dùng rất lớn.
* Những tính năng cốt lõi(core) của Spring là phát triển java Desktop, ứng dụng mobile, java web... Mục tiêu chính là giúp phát triển úng dụng J2EE dễ dàng hơn.
* Kiến trúc, các Module của Spring Framework:



Hình 12. cấu trúc Spring Framework

* Lợi ích của Spring framework:

+ Cho phép lập trình viên sử dụng POJOs giúp không phải làm việc với EJB, úng dụng các luồng chạy, cấu hình đơn giản hơn nhiều.

+ Spring được tổ chức theo kiểu mô đun, số lượng lớp khá nhìều.

+ Spring hỗ trợ sử dụng khá nhiều công nghệ: ORM framework, loging framework...

* Sppring Framework được xây dụng dựa trên 2 nguyên tắc design chính là : Dependency Injection và Aspect Oriented Programming.
* Một số dự án của Spring: Spring MVC, Spring Security, Spring Boot, Spring Mobile, Spring Data...

### **Spring boot là gì**

* Spring boot là một project nằm trên tầng IO Execution( tầng thực thi) của Spring IO Framework.
* Spring boot là một Framework giúp chúng ta thiết lập và phát triển củng như chạy ứng dụng một cách nhanh chóng.
* Spring boot các cấu hình Spring được giảm thiểu tối đa, hỗ trợ các bộ nhúng( embedded container) cho phép úng dụng web chạy độc lập mà không cần khai thác lên web server.
* Các thư viện cần thiết cho Spring boot:



Hình 13. Các thư viện cần thiết

* Lợi ích của String boot :

+ Dễ dàng phát triển ứng dụng dựa trên Spring với java.

+ Giảm thiểu thời gian phát triển và tăng năng suất.

+ Dễ dàng tương tác với các môi trường của Spring nhhuw Spring JDBC, Spring Security, Spring Data...

+ Cung cấp các server nhúng ( Embedded servers) như Tomcat, Jetty... để phát triển và test ứng dụng web dễ dàng.

+ Cung cấp rất nhiều các Plugin để phát triển và test các ứng dụng Spring boot nhanh chóng sử dụng các công cụ build như Maven và Gradle.

#### Viết trên phần mềm nào

Để xây dựng các ứng dụng Spring boot trên IDE eclipse cần phải cài đặt Spring Tool Suite đây là Plugin mở rộng hỗ trợ lập trình Spring trên eclipse.

#### REST API là gì

**API** (**A**pplication **P**rogramming **I**nterface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

**REST** (**RE**presentational **S**tate **T**ransfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource.

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Những phương thức hay hoạt động này thường được gọi là **CRUD** tương ứng với Create, Read, Update, Delete.

## SỬ DỤNG POSTMAN

* Postman là gì

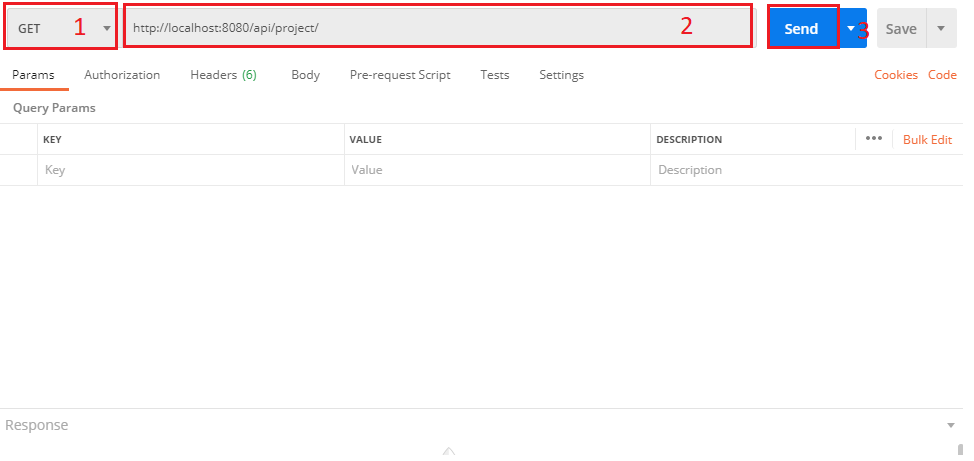
Postman là một công cụ cho phép chúng ta làm việc với API, nhất là REST. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào. Nó hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, …). Bên cạnh đó, nó còn cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần. Các điểm kiểm tra thử nghiệm như xác minh trạng thái phản hồi HTTP thành công có thể được thêm vào mỗi lệnh gọi API giúp đảm bảo phạm vi kiểm tra.

* Các chức năng cơ bản:
* Cho phép gửi HTTP Request với các method GET, POST, PUT, DELETE.
* Cho phép post dữ liệu dưới dạng form (key-value), text, **json**.
* Hiện kết quả trả về dạng text, hình ảnh, XML, JSON.
* Hỗ trợ authorization (Oauth1, 2).
* Cho phép thay đổi header của các request.

Cách test:

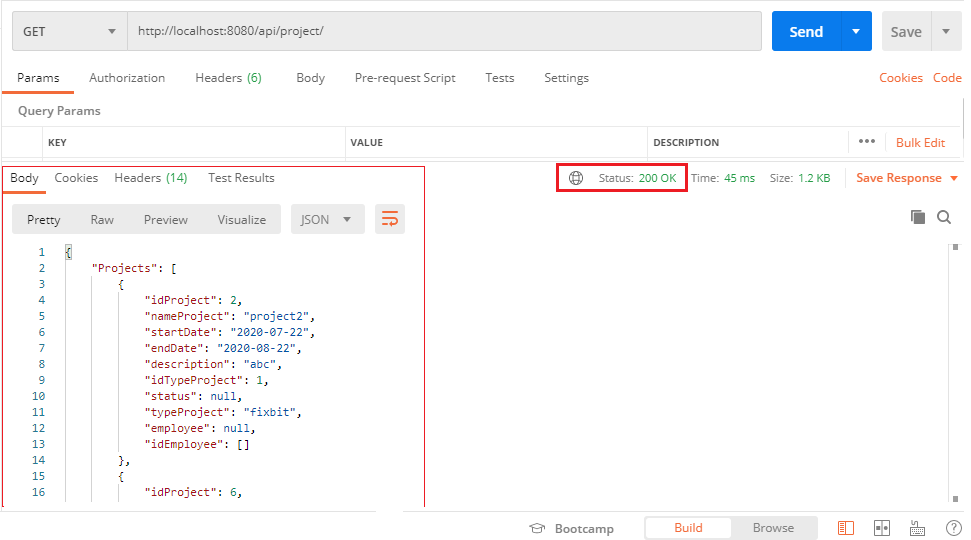
* Request GET được sử dụng để truy vấn thông tin từ truyền vào trên URL. Điều này sẽ không làm thay đổi gì với endpoint.
* Trong Workspace:

1. Thiết lập request HTTP là GET.
2. Trong trường URL yêu cầu nhập vào link.
3. Nhấn nút send.



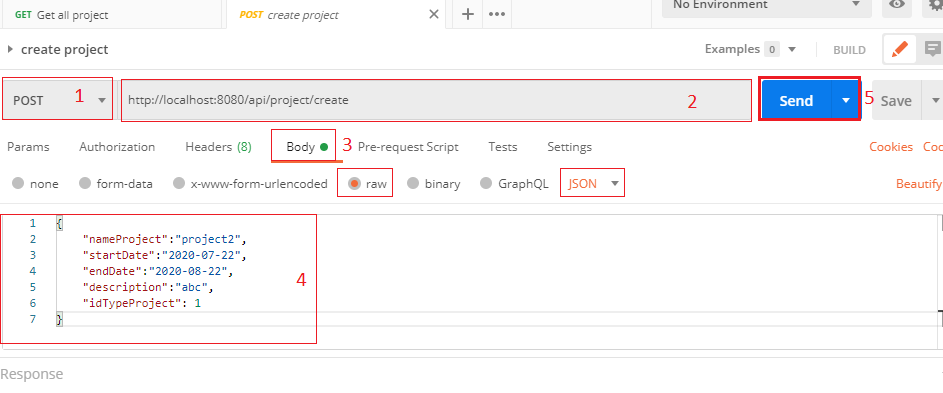
Hình 14. Hướng dẫn sử dụng Postman

1. Sau đó message 200 OK
2. Sẽ hiển thị kết quả trong phần Body.



Hình 15. Hướng dẫn sử dụng Postman

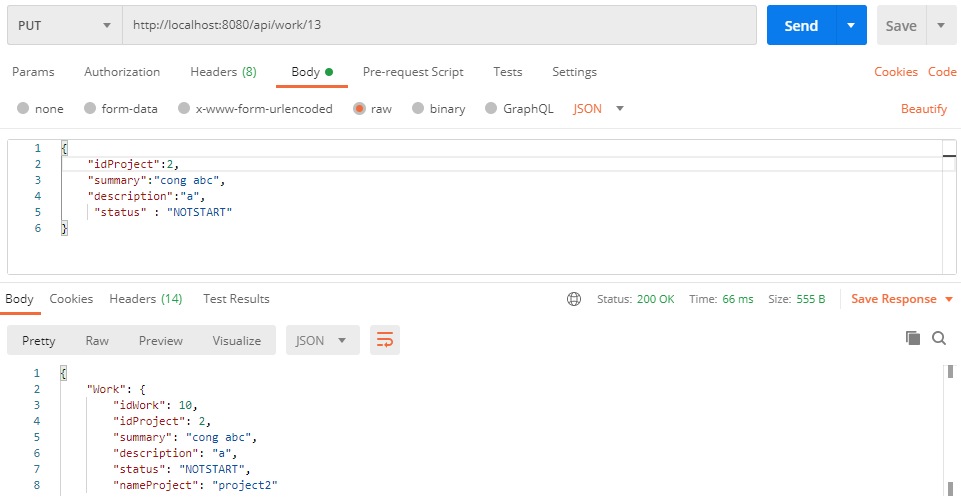
* POST requests khác với GET requests là có thao tác với dữ liệu và người dùng thêm dữ liệu vào endpoint.
  + 1. Thiết lập request HTTP là GET.
    2. Trong trường URL yêu cầu nhập vào link.
    3. Trong tab Body chọn Raw →JSON.
    4. Viết các trường và dữ liệu vào bên trong phần Body.
    5. Nhấn nút send.



Hình 16. Hướng dẫn sử dụng Postman

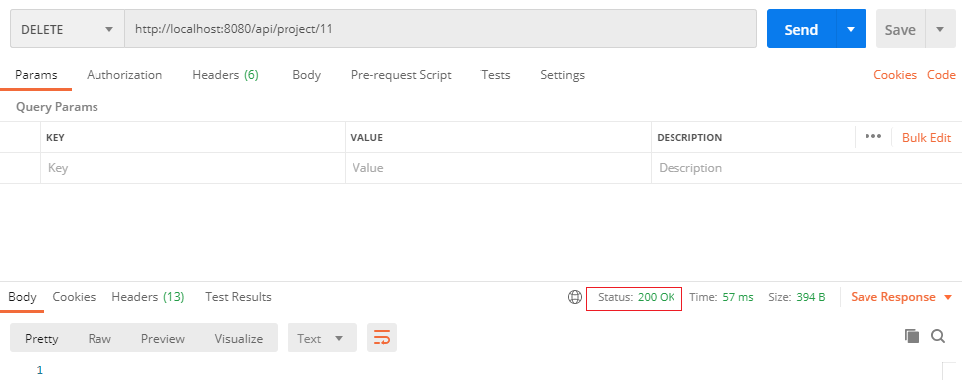
Ta có thể tạo các request khác :

PUT Requests



Hình 17 Hướng dẫn sử dụng Postman

DELETE Requests

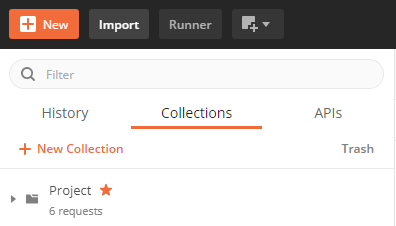


Hình 18. Hướng dẫn sử dụng Postman

* Tham số hóa dữ liệu là một trong những tính năng hữu ích nhất của Postman. Để tạo cùng một request với dữ liệu khác nhau, bạn có thể sử dụng các biến với tham số. Những dữ liệu này có thể từ một tệp dữ liệu hoặc một biến môi trường. Tham số hóa giúp tránh lặp lại các thử nghiệm tương tự và lặp lại có thể được sử dụng để kiểm thử tự động.

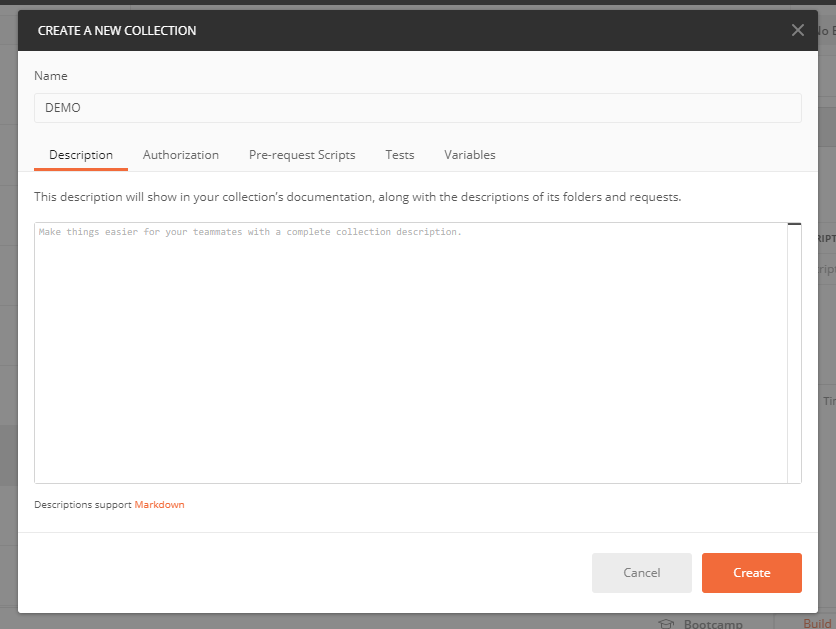
Collection đóng một vai trò quan trọng trong việc tổ chức các bộ thử nghiệm. Nó có thể được import và export để dễ dàng chia sẻ các collection giữa các nhóm. Trong hướng dẫn này, chúng ta sẽ tìm hiểu cách tạo và thực hiện Collection.

1. Chọn Collection → New Collection.



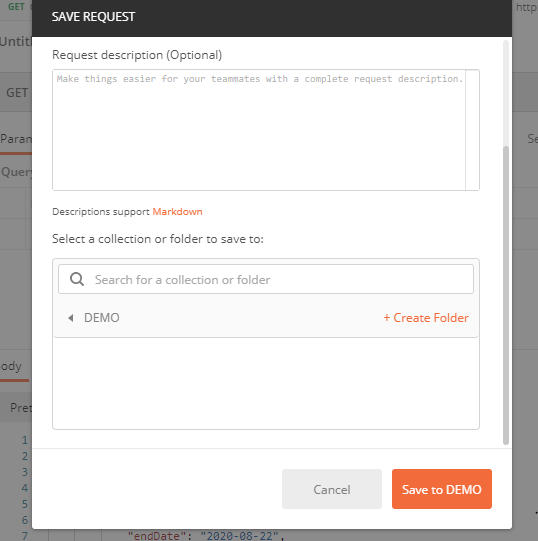
Hình 19. Hướng dẫn sử dụng Postman

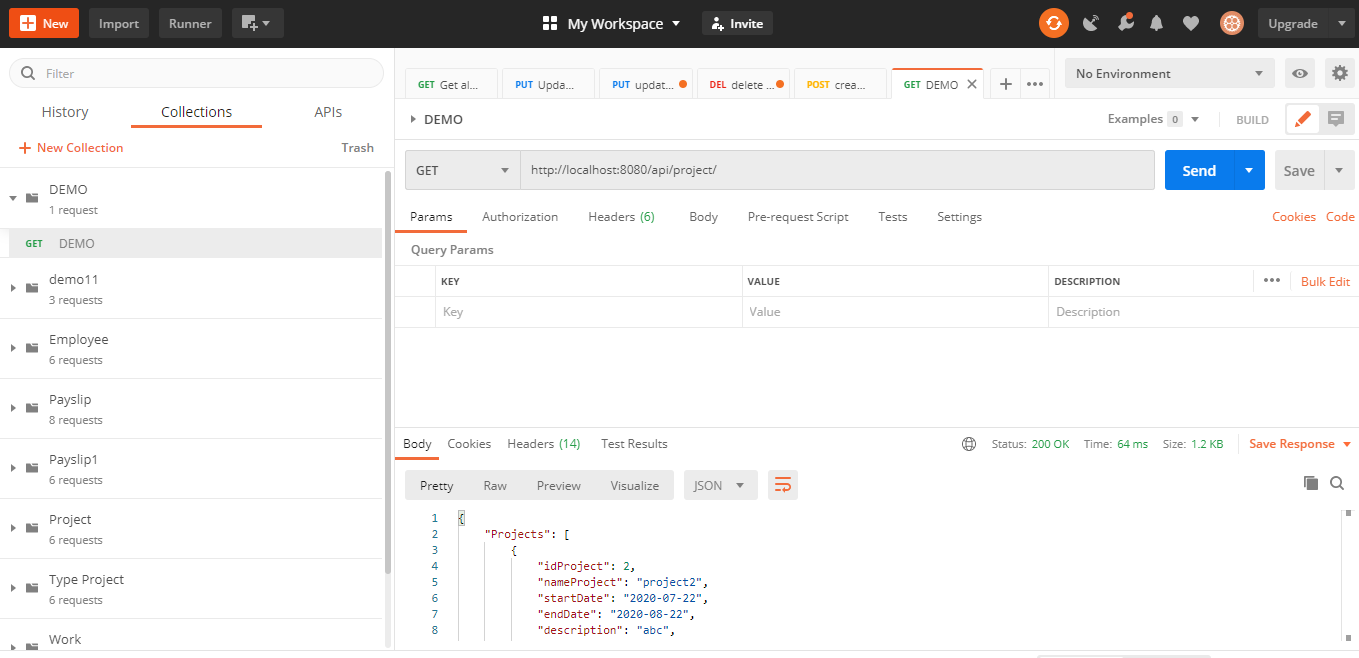
1. Nhập vào tên Collection và mô tả rồi kích nút Create. Một collection lúc này sẽ được tạo.



Hình 20. Hướng dẫn sử dụng Postman

1. Trở lại requets GET phần trước, kích nút **Save**
2. Chọn **DEMO** →Kích nút **Save to DEMO.**





## TÌM HIỂU SPRING ANNOTATION

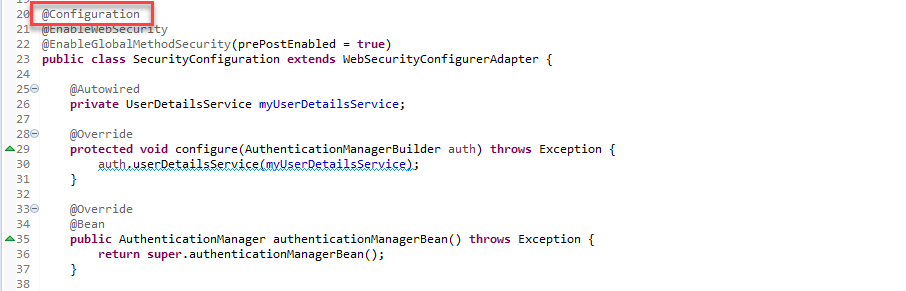
**Annotation**(Chú thích) được sử dụng để cung cấp thông tin dữ liệu cho mã Java.Là thông tin dữ liệu, các **Annotation**không trực tiếp ảnh hưởng đến việc thực hiện các mã của bạn, mặc dù một số loại chú thích thực sự có thể được sử dụng cho mục đích đó. **Annotation**đã được thêm vào Java từ Java 5

**Annotation**được sử dụng cho các mục đích:

1. Chỉ dẫn cho trình biên dịch (Compiler)
2. Chỉ dẫn trong thời điểm xây dựng (Build-time)
3. Chỉ dẫn trong thời gian chạy (Runtime)

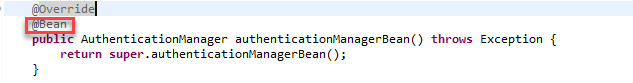
Đây là chức năng quan trọng nhất của Spring Boot, vì Spring Framwork đã xây dựng sẵn cho chúng ta rất nhiều Annotation. Khi server chạy, chúng sẽ quét qua các annotation này và sẽ có thể biết được class nào nên chạy trước, class nào chạy sao và mỗi class sẽ có chức năng gì. Qua đó sẽ giúp server chạy đúng với những gì mà deve mong muốn. Giới thiệu một số annotation sẽ sử dụng trong dự án:

* @Configuration: Được sử dụng để chỉ ra rằng class khai báo sử dụng Annotation @Configuration sẽ khai báo một hoặc nhiều @Bean method trong class đó. Những class khai báo với @Configuration sẽ được Spring Container quản lý và tạo Bean trong lúc chương trình đang chạy.



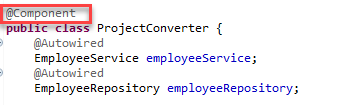
Hình 22. Code dùng trong dự án

* @Bean: Method (phương thức) sử dụng @Bean ở phía trên sản xuất ra đối tượng bean và được quản lý bởi Spring Container. Bean Annotation có thể sử dụng với các tham số như name, initMethod hoặc destroyMethod.



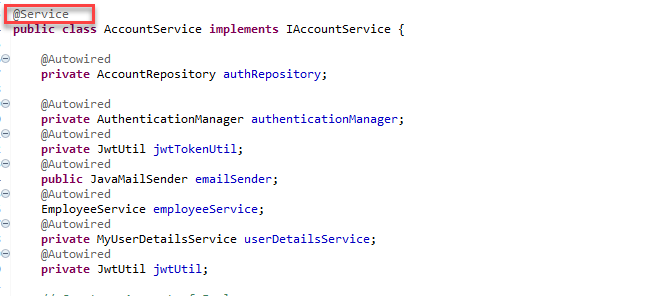
Hình 23. Code dùng trong dự án

* @Component: Khi một class được đánh dấu là Component thì sẽ được tạo thành 1 Bean. Khi Spring Start thì nó quét qua các Annotation có đánh dấu là @Component thì nó sẽ tạo Bean cho class đó. Ví dụ ta có class Contact và ta đánh dấu nó là @Component thì Spring khi đọc qua class này nó sẽ tạo 1 Bean có tên là Contact trong Container của nó. Nếu có class nào dùng thì nó sẽ nhúng Bean này vào. Dùng @Component là để tạo ra một bean.



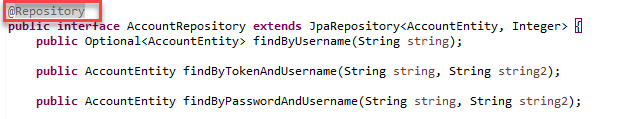
Hình 24. Code dùng trong dự án

* @Service: Nếu một class được đánh dấu là @Service thì nó là kiểu đặc biệt của @Component. Nó được dùng để xử lý các nghiệp vụ của ứng dụng.



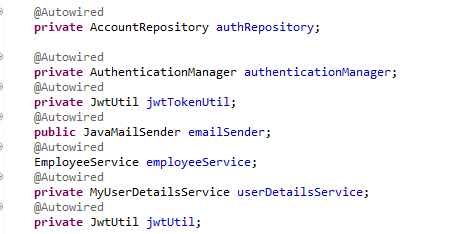
Hình 25. Code dùng trong dự án

* @Repository: Nếu một class được đánh dấu là @Repository thì nó là kiểu đặc biệt của @Component . Nó được sử dụng để nói Bean này dùng để truy cập và thao tác xuống cơ sở dữ liệu.



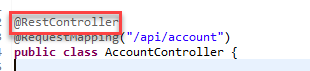
Hình 26. Code dùng trong dự án

* @Autowired: Tự động nhúng các Bean được Spring Container sinh ra vào Class có khai báo @Autowire.



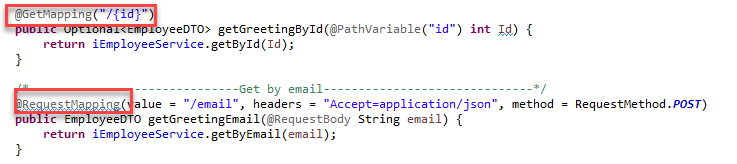
Hình 27. Code dùng trong dự án

* @RestController: Một class được đánh dấu là Controller thì để khai báo Class đó là một Controller và có nhiệm vụ Mapping Request trên URL vào các method tương ứng trong Controller. Đây chính là nơi chứa các api endpoint của server.



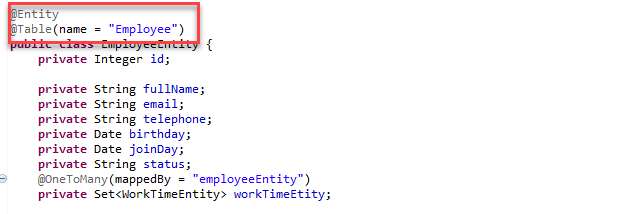
Hình 28. Code dùng trong dự án

* CRUD annotation: Bao gồm @PutMapping, @GetMapping, @PostMapping, @DeleteMapping, @RequestMapping . Các annotation này được dùng để định nghĩa các phương thức http giao tiếp giữa client và server.



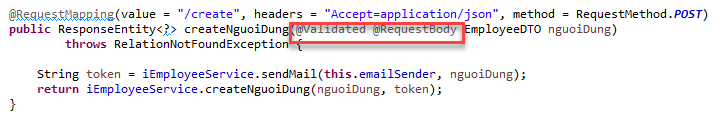
HÌnh 29. Code dùng trong dự án

* @Entity, @Table: được dùng để khai báo một bảng dữ liệu trong Spring Boot. Đây là nơi chứa và nhận dữ liệu từ database cũng như client khi deploy server.



Hình 30. Code dùng trong dự án

* Một số annotation khác: @Validation dùng để ràng buộc dữ liệu; @RequestParam ,@RequestBody, @RequestHeader dùng để nhận dữ liệu từ client,….



Hình 31. Code dùng trong dự án

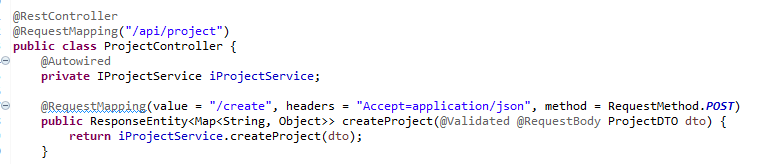
## TÌM HIỂU SPRING CONTROLLER

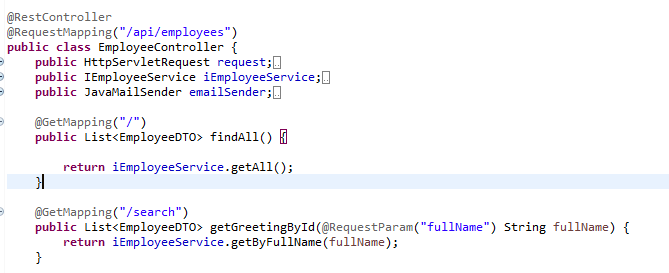
Spring Controller annotation thường được sử dụng kết hợp các phương thức xử lí chú thích dựa trên RequestMapping annotation. Spring Controller annotation chỉ có thế áp dụng lên các class. Nó dùng để đánh dấu class là xử lí web requests.

Spring @RestController là annotation tiện lợi được kết từ @Controller và @ResposeBody. @RestController là một controller đặc biệt được sử dụng các dịch vụ RESTful webservice.

Khi đặt một Controller @RestController thì mọi method controller sẽ được thừa hưởng annotation @ResponseBody, response data trong controller này sẽ được trả về dưới dạng message.

Các đối tượng mà @RestController trả về rất đa dạng, có thể trả về các dữ liệu như: Object, List, Map...Spring Boot sẽ tự động convert hết chúng thành json.





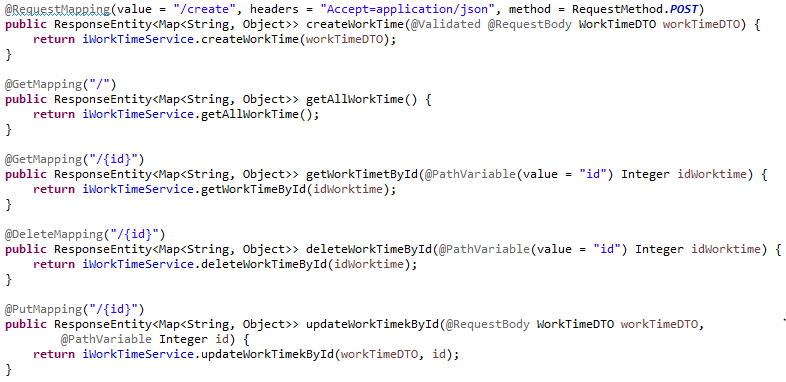
Hình 32. Code dùng trong dự án

@RequestMapping trong Spring Boot

@RequestMapping là annotation khai báo ở cấp độ class và method, dùng để đăng kí và ánh xạ các request tới một method của controller dựa vào các thuộc tính như URL, HTTP method, requests parameters, headers, media types. Dùng @RequestMapping ở cấp độ class thì các method trong class controller củng được thừa hưởng append với nhau tạo ra endpoint cuối cùng.

Khi một request đến, dispatcher servlet sẽ chuyển tiếp request xuống handler mapping, sau đó dựa vào thông tin request kết hợp với @RequestMapping đã đăng ký trước đó để điều hướng tới một method controller cụ thể.

Spring cung cấp các annotation GetMapping, PostMapping, DeleteMapping, PutMapping giúp rút ngắn việc khái báo hơn.



Hình 33. Code dùng trong dự án

@GetMapping là cấp độ method dùng để ánh xạ các yêu cầu HTTP GET tới một phương thức cụ thể.

@PostMapping(@RequestMapping(method = RequestMethod.POST)) dùng để tạo mới

@PutMapping dùng để update dữ liệu

@DeleteMapping dùng để xóa dữ liệu

## TÌM HIỂU SPRING DATA

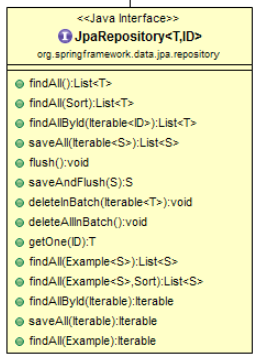
Spring Data là một module của Srinng Framework. Mục đích của Spring Data là giảm thiểu việc thực hiện quá nhiều bước để có thể implement được JPA. Spring Data JPA là một phần của Spirng Data và nó hỗ trợ Hibernate 5, OpenJPA 2.4 và EclipseLink 2.6.1

Các tính năng cơ bản:

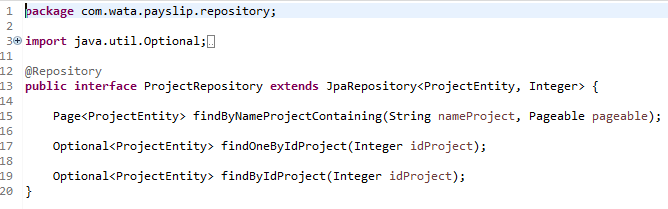
* Hỗ trợ xây dựng repositoties dựa trên Spring và JPA.
* Hỗ trợ phân trang và thực hiện Query động.
* Hỗ trợ XML Mapping.
* Cấu hình java Config bằng việc sử dụng @EnableJpaRepositories.

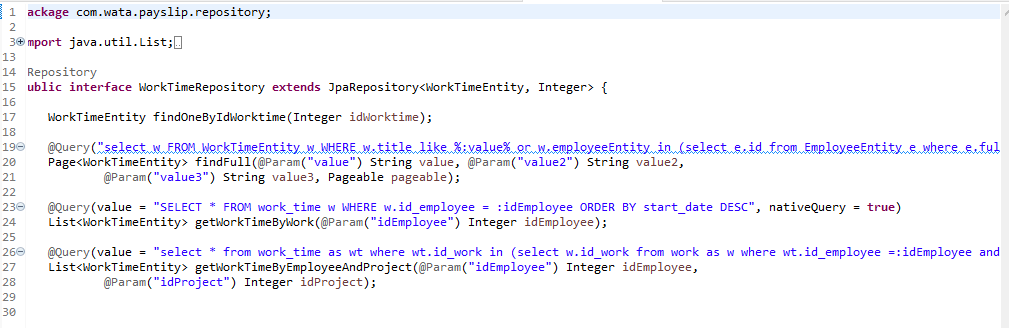
Spring Data JPA project cung cấp 2 Interface:

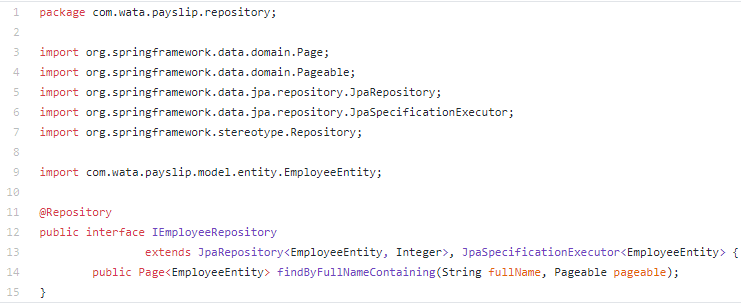
* JPAREPOSITORY<T,ID extends Serializable> interface
* JpaSepecificatiionExecutor interface

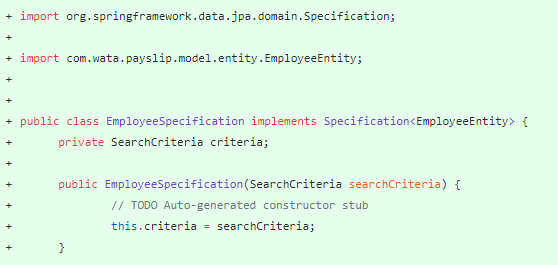


Hình 35. Method trong JPA









Hình 36. Sử dụng JPA

## TÌM HIỂU SPRING SECURITY

### **Giới thiệu**

Đây là một framework được xây dựng nhằm xác thực và phân quyền cho server. Ngoài ra còn cung cấp các nhiều service khác giúp cấu hình và bảo vệ server khỏi những attack không mong muốn, đảm bảo an toàn cho server.

## Cơ chế hoạt động

Khi một request được gửi đến server, Spring security sẽ dựng một middle man đứng giữa chặn lại request đó và xác thực request có hợp lệ hay không tùy theo những gì mà dev đã cấu hình, nếu hợp lệ thì request sẽ có thể truy cập vào api , nếu không thì sẽ trả lại trạng thái 403(Access Denied).

Ngoài ra còn có thể bảo mật trên lớp controller, có nghĩa là khi request qua được tầng middle man của Spring secure thì sẽ bị chặn lại ở endpoint api. Khi đó server sẽ xác thực request đó có đủ quyền hay không dựa vào hệ thống phân quyền mà chúng ta cấu hình. Nếu đủ thì sẽ truy cập được api, nếu không vẫn sẽ trả lại trạng thái 403 (Access Denied) .

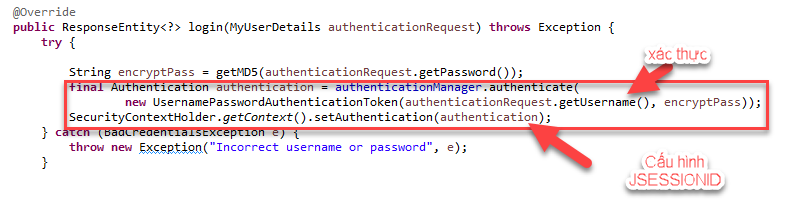
## Xác thực người dùng ( Authentication )

Để truy cập vào các nguồn dữ liệu (Resource) được bảo mật bởi **Basic Authentication**, người dùng phải gửi một **request** và trong **request** đó có chứa thông tin **username/password** được đính kèm trên **Header**.

Sau đó, Interface AuthenticateManager sẽ có nhiệm vụ xác thực thông tin và nếu đúng, sẽ gửi về một JSESSIONID cookies.Những lần truy cập sau , người dùng sẽ gửi cùng với cookies này để có thể truy cập những resource đã được bảo mật.

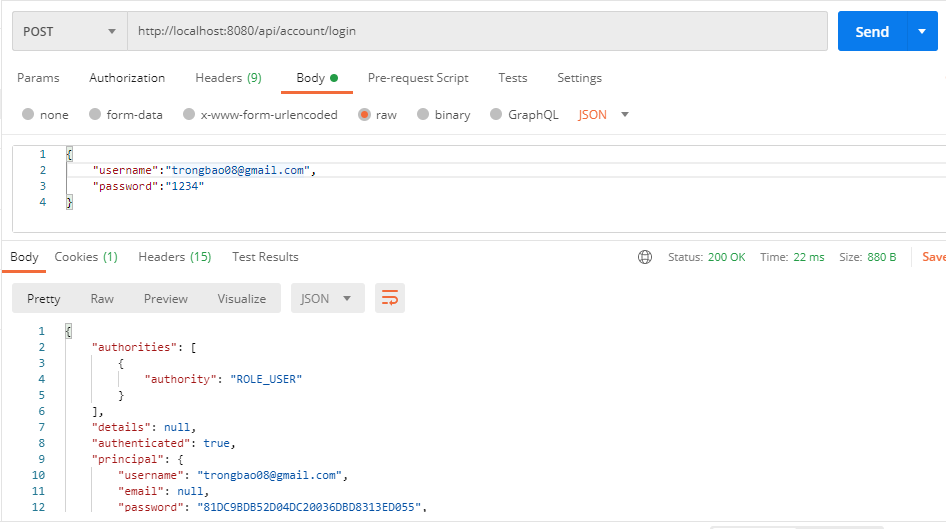
Chức năng login:

Code:



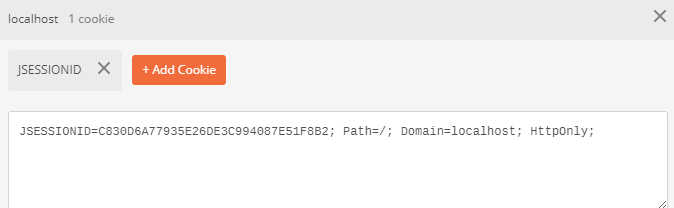
Hình 37. Code dùng trong dự án

Kiểm tra login bằng PostMan



Hình 38. Đăng nhập

Cookies nhận được:

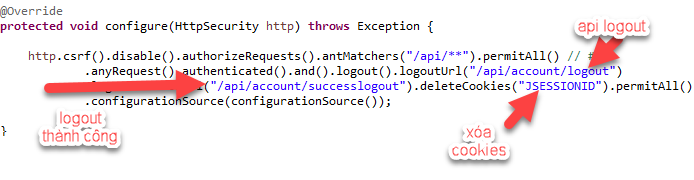


Hình 39. Kết quả nhận được

Như vậy là đã xây dựng xong chức năng login và test thành công. Như đã nói ở trên, mỗi lần truy cập sau vào những tài nguyên được cấu hình yêu cầu xác thực, chúng ta cần phải gửi chung với JSESSIONID cookies. Nếu không có sẽ trả về status 403(Access Denied).

Chức năng Logout: chức năng này có phần đơn giản hơn chức năng login bởi vì người dùng truy cập vào api logout , server sẽ bắt được JSESSIONID cookies, server sẽ chỉ cần xóa JSESSIONID cookies đi và trả về status 200. Như vậy những lần truy cập sau, JSESSIONID cookies này sẽ không thể sử dụng được nữa, người dùng sẽ cần phải đăng nhập để có thể nhận JSESSIONID cookies mới.

Code:



Hình 40. Code dùng trong dự án

Trên đây là 2 chức năng login và logout sử dụng Spring security được xây dựng trong dự án có nhiệm vụ quản lý xác thực người dùng.

* 1. **Xây dựng phân quyền ( Authorization )**

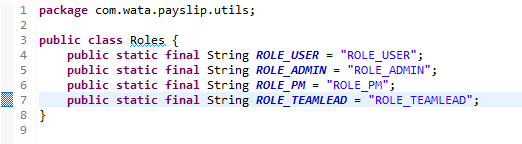
Dự án đã xây dựng thành công chức năng xác thực người dùng, nhưng có một câu hỏi mới: Làm thế nào để có thể kiểm soát những người dùng hợp lệ chỉ có thể truy cập vào những tài nguyên cho phép, những tài nguyên còn lại cần nhiều quyền hơn chứ không còn đơn giản là xác thực ? Để giải quyết yêu cầu này, Spring Security mang đến cho chúng hệ thống phân quyền ( Authorization ) giúp chúng ta kiểm soát quyền truy cập tài nguyên tùy theo mỗi nhóm người dùng có vai trò riêng biệt.

Trong dự án , mình xây dựng 3 vai trò( Roles) có chức năng khác nhau:

ROLE\_USER : Người dùng bình thường với quyền hạn chế , không thể thay đổi dữ liệu hệ thống, chỉ có thể xem và cập nhật thông tin cá nhân

ROLE\_PM : Project Manager có nhiều quyền hơn ROLE\_USER, role này sẽ có nhiệm vụ quản lí thông tin dự án,chia dự án thành nhiều công việc, phân công nhân viên và một số chức năng khác.

ROLE\_ADMIN: Admin có quyền cao nhất hệ thống, có nhiệm vụ quản lý vai trò của nhân viên và project manager cũng như quản trị hệ thống, giúp cho hệ thống hoạt động một cách hiệu quả nhất.



Hình 41. Minh họa một số vai trò trong hệ thống

antMatchers(): khai báo đường dẫn của request  
permitAll(): cho phép tất cả các user đều được phép truy cập.  
hasRole(roleName): chỉ cho phép các user có GrantedAuthority là ROLE\_roleName mới được phép truy cập



HÌnh 42. Cấu hình Authorization

Ngoài ra còn một số phương thức khác, chúng ta có thể tham khảo thêm tại

<https://docs.spring.io/spring-security/site/docs/current/reference/html5/#el-common-built-in>

## Cài đặt CorConfigs

Như vậy bảo mật dự án về cơ bản đã hoàn thành, đã có thể xác thực và phân quyền dựa trên mỗi request, khi sử dụng postman để test thì dự án chạy tốt. Tuy nhiên , khi xây dựng phía client đã xảy ra một vấn đề nghiêm trọng : Chặn truy cập server từ một domain khác, tức mọi server khác khi truy cập vào api thì đều bị chặn bởi CorConfigs.

Cor là gì?

**CORS** là một cơ chế cho phép nhiều tài nguyên khác nhau (fonts, Javascript, v.v…) của một trang web có thể được truy vấn từ domain khác với domain của trang đó. **CORS** là viết tắt của từ Cross-origin resource sharing.

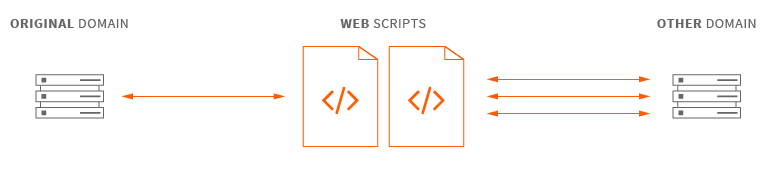
CORS được sinh ra là vì [same-origin policy](https://www.w3.org/Security/wiki/Same_Origin_Policy), là một chính sách liên quan đến bảo mật được cài đặt vào toàn bộ các trình duyệt hiện nay. Chính sách này ngăn chặn việc truy cập tài nguyên của các domain khác một cách vô tội vạ.

Ta có ví dụ một kịch bản như sau:

* Bạn truy cập một trang web có mã độc. Trang web đó sử dụng Javascript để truy cập tin nhắn Facebook của bạn ở địa chỉ https://facebook.com/messages.
* Nếu bạn đã đăng nhập Facebook từ trước rồi. Nếu không có same-origin policy, trang web độc hại kia có thể thoải mái lấy dữ liệu của bạn và bất cứ điều gì chúng muốn.

Same-origin policy chính là để ngăn chặn những kịch bản như trên để bảo vệ người dùng, giúp an toàn hơn khi lướt web.

**CORS** sử dụng các HTTP header để “thông báo” cho trình duyệt rằng, một ứng dụng web chạy ở origin này (thường là domain này) có thể truy cập được các tài nguyên ở origin khác (domain khác).

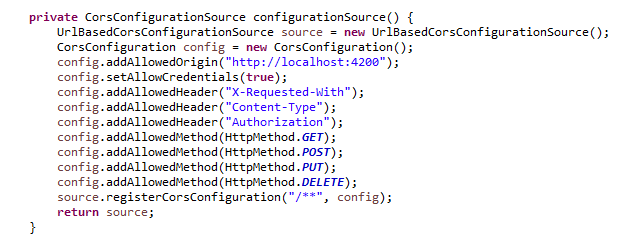


Hình 43. Minh họa cách Cors hoạt động

Một hiểu lầm khá phổ biến, nhất là với các lập trình viên mới làm việc với API lại được làm việc với API của các hãng lớn, tài liệu đầy đủ, đó là cho rằng CORS là công việc của frontend. Nhưng thực ra CORS hoàn toàn là [công việc của backend](https://topdev.vn/blog/lo-trinh-tro-thanh-backend-developer/).

**Cấu hình CorsConfig**

Trong Spring Boot, Cors cũng sẽ được cấu hình trong SecurityConfig. Có rất nhiều cách để cấu hình cors, nhưng để đảm bảo dự án có thể chạy ở mọi nơi mà không cần phải setup lại cors config quá nhiều, chúng ta config như sau :



Hình 44. Code dùng trong dự án

Hình trên được hiểu như sau :

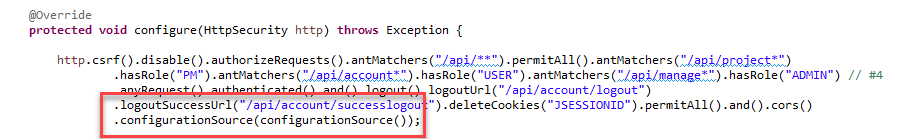
Config.addAllowedOrigin: Danh sách những domain được phép truy cập.

Config.addAllowedHeader: những phương thức trên Header được cho phép.

Config.addAllowedMethod: Những phương thức http được cho phép

Source.registerCorsConfiguration: Danh sách API được áp dụng cấu hình này.

Cuối cùng, đưa những cài đặt này vào Security Config:



Như vậy , theo cấu hình Cors trên, từ giờ chỉ có domain <http://localhost:4200> được phép truy cập và sử dụng API từ server, tất cả các domain khác đều sẽ bị chặn. Điều này sẽ hạn chế những truy cập vô tội vạ từ những domain khác.

## KẾT QUẢ THỰC NGHIỆM

**Chức năng đã xây dựng**

* Đăng ký nhân viên mới
* Đăng nhập , đăng xuất
* Quên mật khẩu , cập nhật mật khẩu
* Phân quyền cho nhân viên
* Quản lý nhân viên
* Quản lý dự án
* Phân chia dự án thành các task
* Phân công nhân viên
* Chấm công cho nhân viên
* Quản lý thời gian công việc trong dự án

**Kiến thức đạt được:**

Hiểu cách thức xây dựng một API hoàn chỉnh, nguyên lý hoạt động cũng như các vấn đề về bảo mật.

Học được Framework Spring Boot, cách xây dựng một dự án hoàn chỉnh bằng Spring Boot.Ngoài ra còn hiểu thêm về cách thức giao tiếp thông qua http.

Hiểu được cách thức hoạt động của một team xây dựng dự án trong thực tế

## Kết luận chương 2

Sau khi thực tập tại doanh nghiệp, sinh viên đã học được rất nhiều điều, không chỉ kiến thức thực tiễn, củng cố kiến thức sẵn có mà còn học được cả những kỹ năng mềm cần thiết để giúp công việc hoàn thành tốt hơn, làm việc hiệu quả hơn, tư duy logic tốt hơn qua việc tham gia vào các dự án thực tế.

# CHƯƠNG 3: KẾT QUẢ THỰC TẬP

Qua quá trình thực tập tại doanh nghiệp, sinh viên đã thu được những kết quả:

* Ứng dụng được những kiến thức nền tảng đã được học ở trường vào môi trường doanh nghiệp.
* Được trải nghiệm, học hỏi kỹ năng mềm khi tham gia vào những dự án:
* Kỹ năng giao tiếp với team
* Kỹ năng tư duy logic
* Kỹ năng tư duy phản biện
* Kỹ năng thích nghi
* Kỹ năng lắng nghe và học tập từ những lời phê bình
* Kỹ năng quản lý thời gian
* Kỹ năng sắp xếp công việc
* Kỹ năng làm việc với áp lực
* Kỹ năng giải quyết vấn đề
* Kỹ năng tự nghiên cứu
* Học hỏi được nhiều kiến thức, kinh nghiệm từ CGDN.
* Chủ động hơn, mong muốn trau dồi thêm những thứ mình chưa có hoặc còn yếu, hòa đồng với mọi người.

**BẢNG GHI NHẬN KẾT QUẢ THỰC TẬP HÀNG TUẦN**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC SÀI GÒN  **KHOA CÔNG NGHỆ THÔNG TIN** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  *Tp. Hồ Chí Minh, ngày 02 tháng 09 năm 2020* |

Mẫu 06\_TTTN\_k.CNTT

**BẢNG GHI NHẬN KẾT QUẢ THỰC TẬP HÀNG TUẦN**

|  |  |  |
| --- | --- | --- |
| **Một số thông tin liên hệ** | | |
| Họ và tên: Thiều Trọng Bảo |  |
| Ngày sinh: 09-09-1998 |  |
| Mã số sinh viên: 3116410006 |  |
| Lớp: DCT1165 |  |
| Ngành học: Công Nghệ Thông Tin |  |
| Email: [trongbao0909@gmail.com](mailto:trongbao0909@gmail.com) |  |
| Điện thoại: 0938352418 |  |
|  |  |
| Chuyên gia doanh nghiệp: Vương Chí Quang |  |
| Email: [quang.vuong@watasolutions.com](mailto:quang.vuong@watasolutions.com) |  |
| Điện thoại: 0908 868 527 |  |
|  |  |
| Giảng viên hướng dẫn: Nguyễn Quốc Huy |  |
| Email: [nqhuy@sgu.edu.vn](mailto:nqhuy@sgu.edu.vn) |  |
| Điện thoại: 0903714586 |  |

|  |  |  |
| --- | --- | --- |
| **Tuần** | **Nội dung thực tập**  **(do chuyên gia của doanh nghiệp giao)** | **Kết quả thực tập**  **(do chuyên gia của doanh nghiệp đánh giá)** |
| 1  Từ ngày 22/07/2020  đến ngày 24/07/2020 | * Meeting : giới thiệu về công ty, chính sách và hoạt động của công ty. * Nhận vị trí thực tập: Java Web, cài đặt môi trường cần thiết. * Nhận đề tài: Payslip, Meeting phổ biến đề tài. * Tìm hiểu tổng quan Spring Boot trong Spring Framework. Tìm hiểu Gitlab | Hoàn thành |
| 2  Từ ngày 27/07/2020  đến ngày 31/07/2020 | * Meeting phân công việc , các kĩ thuật sử dụng, các thành phần, chức năng tổng quan của ứng dụng. * Cài đặt spring boot version 2.3.1. * Thực hiện kết nối database sql server 2012, sử dụng mssql-jdbc version 7.4.1.jre8. * Cài đặt Tomcat Server | Hoàn thành tốt |
| 3  Từ ngày 03/08/2020  đến ngày 07/08/2020 | * Training/Sharing: Kĩ năng làm việc nhóm theo mô hình scrum. * Xây dựng REST API cho employee. * Tạo dữ liệu database. * Tìm hiểu công cụ Postman. * Tìm hiểu phương pháp validate dữ liệu đầu vào. | Hoàn thành tốt |
| 4  Từ ngày 10/08/2020  đến ngày 14/08/2020 | * Sử dụng Sourcetree để quản lí project. * Meeting: Tối ưu hóa và cấu trúc lại package trong project. * Tìm hiểu và sử dụng Spring JPA. * Tham dự open meeting về Docker. * Xây dựng Sping Security cơ bản để phân quyền truy cập. | Hoàn thành tốt |
| 5  Từ ngày 17/08/2020  đến ngày 21/08/2020 | * Tạo API cho account(register, login, logout, forgor password, update info) * Tìm hiểu sử dụng anotation @query trong Spring Jpa. * Xây dựng Authentication và Authorization bằng Spring Security * Set up quyền truy cập server bằng Spring Security(Cors Config). * Hoàn thiện code. | Hoàn thành xuất sắc  - Tìm hiểu khá tốt về Spring Security, Authenticate và Authorize thành công cho hệ thống.  - Có khả năng tìm hiểu và giải quyết vấn đề tốt. |
| 6  Từ ngày 24/08/2020  đến ngày 28/08/2020 | * Sửa lỗi và cập nhật lại dữ liệu output. * Tạo API phân công công việc cho nhân viên. * Cài đặt vai trò và quyền của người dùng(Role) ở mức Controller. * Mã hóa và lưu mật khẩu bằng thuật toán MD5. * Thêm thuộc tính trạng thái vào đối tượng nhân viên và project. | Hoàn thành tốt |
| 7  Từ ngày 31/08/2020  đến ngày 04/09/2020 | * Viết báo cáo | Hoàn thành tốt |
| 8  Từ ngày 07/09/2020  đến ngày 11/09/2020 | * Viết báo cáo | Hoàn thành tốt |

**Chuyên gia doanh nghiệp hướng dẫn thực tập**

(Ký tên và ghi họ tên)

Vương Chí Quang

**Ghi chú:**

-**Chuyên gia doanh nghiệp** ghi nhận kết quả thực tập của **sinh viên** theo tuần và gởi qua email cho **giảng viên hướng dẫn** khi kết thúc các tuần 3,6 của đợt thực tập. BẢNG GHI NHẬN KẾT QUẢ THỰC TẬP TỐT NGHIỆP HÀNG TUẦN này là một trong những hồ sơ kèm theo quyển báo cáo thực tập tốt nghiệp.

-Cột **Kết quả thực tập**, chuyên gia doanh nghiệp có thể ghi *Hoàn thành tốt*, *Hoàn thành*, *Không đạt* hoặc có thể ghi nhận chi tiết hơn.

# 

**BẢNG ĐÁNH GIÁ QUÁ TRÌNH THỰC TẬP TỐT NGHIỆP**

|  |  |
| --- | --- |
| **ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH** **TRƯỜNG ĐẠI HỌC SÀI GÒN**  **KHOA CNTT** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc** |

Mẫu 07\_TTTN\_k.CNTT

**BẢNG ĐÁNH GIÁ QUÁ TRÌNH THỰC TẬP TỐT NGHIỆP**

**(do chuyên gia doanh nghiệp đánh giá).**

Họ và tên sinh viên: Thiều Trọng Bảo

Ngày sinh : 09/09/1998

Mã số sinh viên: 3116410006

Lớp: DCT1165

Thời gian thực tập : 22/07/2020 – 13/07/2020

Doanh nghiệp thực tập : WATA SOLUTIONS

Địa chỉ doanh nghiệp: 39B Trường Sơn , phường 2, quận Tân Bình, tp. Hồ Chí Minh

Chuyên gia doanh nghiệp hướng dẫn: Vương Chí Quang

**I. ĐÁNH GIÁ VỀ QUÁ TRÌNH THỰC TẬP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Nội dung đánh giá** | **ĐIỂM** | | |
| **0** | **0.5** | **1** |
| 1 | Khả năng thực hành |  |  | **X** |
| 2 | Khả năng làm việc nhóm |  |  | **X** |
| 3 | Tính thân thiện |  |  | **X** |
| 4 | Tính năng động |  |  | **X** |
| 5 | Tính thần sáng tạo |  |  | **X** |
| 6 | Chấp hành nội quy cơ quan |  |  | **X** |
| 7 | Giờ giấc làm việc |  |  | **X** |
| 8 | Phương pháp làm việc |  |  | **X** |
| 9 | Khối lượng công việc |  |  | **X** |
| 10 | Báo cáo thực tập tốt nghiệp |  |  | **X** |

(theo thang điểm 10).

**II. CÁC ĐÁNH GIÁ KHÁC:** .............................................................................................................................................................................................................................................................................................................................................................................................................................................**III. KẾT QUẢ TỔNG HỢP:**

**Điểm tổng cộng : 10**

|  |  |
| --- | --- |
| **XÁC NHẬN CỦA DOANH NGHIỆP**  (đóng mộc tròn của doanh nghiệp, họ tên, ký tên) | **Chuyên gia hướng dẫn**  *(Ký và ghi họ tên)* |

Vương Chí Quang

**PHIẾU ĐÁNH GIÁ KẾT QUẢ THỰC TẬP TỐT NGHIỆP**

(Do giảng viên hướng dẫn đánh giá) mẫu 8

Mẫu 08\_TTTN\_k.CNTT

|  |  |
| --- | --- |
| **ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH** **TRƯỜNG ĐẠI HỌC SÀI GÒN**  **KHOA CNTT** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc** |

**PHIẾU ĐÁNH GIÁ KẾT QUẢ THỰC TẬP TỐT NGHIỆP**

**(do giảng viên hướng dẫn đánh giá)**

Họ và tên sinh viên: Thiều Trọng Bảo

Ngày sinh: 09/09/1998

Mã số sinh viên: 3116410006

Lớp: DCT1165

Thời gian thực tập: 22/07/2020 – 20/09/2020

Doanh nghiệp thực tập: WATA SOLUTIONS

Địa chỉ doanh nghiệp: 247/8B Hoàng Hoa Thám, Phường 05, Quận Phú Nhuận, TP Hồ Chí Minh

Chuyên gia doanh nghiệp hướng dẫn: Vương Chí Quang

10

**I. ĐIỂM CỦA CHUYÊN GIA DOANH NGHIỆP :**

(thang điểm 10)

**II. ĐIỂM CỦA GIẢNG VIÊN HƯỚNG DẪN:**

(thang điểm 10)

**II. ĐIỂM TỔNG KẾT:**

(trung bình cộng 2 cột điểm trên, thang điểm 10)

**Xếp loại :**

*TP Hồ Chí Minh ngày tháng năm*

**Giảng viên hướng dẫn**

***GHI CHÚ :***

*Xếp loại được tính như sau:*

*Xếp loại Xuất sắc nếu điểm tổng hợp từ 9 đến 10*

*Xếp loại Giỏi nếu điểm tổng hợp từ 8 đến cận 9*

*Xếp loại Khá nếu điểm tổng hợp từ 7 đến cận 8*

*Xếp loại TB Khá nếu điểm tổng hợp từ 6 đến cận 7*

*Xếp loại TB nếu điểm tổng hợp từ 5 đến cận 6*

*Xếp loại Không đạt nếu điểm tổng hợp từ 0 đến cận 5*

**Kết luận chương 3**

Những kết quả sinh viên đạt được là nhờ quá trình học tập và làm việc tại doanh nghiệp, giúp sinh viên phát huy tối đa khả năng của mình, ứng dụng được những kiến thức trong sách vở vào thực tiễn ngoài đời, học hỏi, trau dồi được nhiều điều mà chỉ có ở môi trường doanh nghiệp mới có.

# CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ

Những nội dung sinh viên đã học hỏi được qua đợt thực tập:

* Nắm được quy trình, nghiệp vụ của doanh nghiệp
* Tích lũy được kiến thức, kinh nghiệm, kỹ năng mềm cần thiết
* Học hỏi được cách giải quyết các vấn đề ở thế giới thực qua các dự án
* Nâng cao kỹ năng giao tiếp, phân chia công việc hiệu quả

Kiến nghị của sinh viên:

Công tác thực tập nhờ sự kết hợp của doanh nghiệp Kyanon Digital và trường Đại học Sài Gòn, rất mong nhà trường và doanh nghiệp tiếp tục phát huy, giữ vững mối quan hệ để giúp đỡ các sinh viên sau này hoàn thành báo cáo thực tập tốt nghiệp.

# TÀI LIỆU THAM KHẢO

1. <https://spring.io/guides#tutorials>
2. <https://loda.me/spring-boot/>.
3. <http://zetcode.com/springboot/>
4. <https://www.javaguides.net/2019/01/spring-boot-microsoft-sql-server-jpa-hibernate-crud-restful-api-tutorial.html>
5. <https://kipalog.com/>